

Neutron Source Facility Simulator (NSFS)

Nuclear Engineering Division

About Argonne National Laboratory

Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

DOCUMENT AVAILABILITY

Online Access: U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via DOE's SciTech Connect (<http://www.osti.gov/scitech/>)

**Reports not in digital format may be purchased by the public from the
National Technical Information Service (NTIS):**

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd Alexandria, VA 22312 www.ntis.gov
Phone: (800) 553-NTIS (6847) or (703) 605-6000
Fax: (703) 605-6900
Email: orders@ntis.gov

**Reports not in digital format are available to DOE and DOE contractors from the
Office of Scientific and Technical Information (OSTI):**

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
www.osti.gov
Phone: (865) 576-8401
Fax: (865) 576-5728

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

Neutron Source Facility Simulator (NSFS)

Prepared by

Austin L. Grelle, Yan Cao, Yousry Gohar, Young Park, and Tom Wei

Nuclear Engineering Division, Argonne National Laboratory

March 2014

NEUTRON SOURCE FACILITY SIMULATOR

NSFS



Abstract

The Neutron Source Facility Simulator (NSFS) was developed to demonstrate the different operating modes of the Neutron Source Facility. The system parameters are changed and the component responses are accelerated for demonstration purposes and training sessions. NSFS shows the pressure, mass flow, and temperature values within the three coolant loops of the facility for training purposes. In the simulation, the operator or supervisor of the training session can modify the operating parameters of the different components within the coolant loops, such as valves or pumps, and witness the transients that occur in the system through the Graphical User Interface (GUI) of NSFS.

Glossary

GUI – *Graphical User Interface*:

A program draws elements to the screen and receive input from the user.

NSFS – *Neutron Source Facility Simulator*:

The acronym describes the simulator.

CSS – *Control System Studio*:

It displays the plant graphics to the screen.

EPICS – *Experimental Physics and Industrial Control System*:

It is utilized to communicate between the simulator and the control system graphics.

Table of Contents

Abstract	i
Glossary	ii
List of Figures	iv
List of Tables	vi
Introduction	1
Hardware Requirements	1
Prerequisites	4
Java SE Development Kit Version 7 – 64-bit	4
MinGW	4
Strawberry Perl 5.16.3.1 – 32bit	5
7-Zip (<i>optional</i>)	5
Python 2.7.5 32-bit	5
Installation / Set-up	7
Installing EPICS Base	7
Installing the Plant Model	13
Installing CSS	13
Finalizing Installation: Combined Operating Environment	17
Getting Started	18
Quick Start Guide	18
Individual Operation of NSFS Component Software	19
Running EPICS	20
Running the Plant Model	21
Running CSS	23
Appendix I: Screenshots	38

List of Figures

Figure 1:	The option of running the entire NSFS suite on a single workstation that the operator then controls.....	2
Figure 2:	The option of running the different components of NSFS on separate systems, where the operator controls the simulator from CPU #1.	2
Figure 3:	The option of running CSS on several different displays (potentially on different computers for each display) while controlling the system from a central workstation utilized by the operator.	3
Figure 4:	The option of running the Plant Model on a different, faster computer than the one that the operator utilizes.	4
Figure 5:	Initially setting the workspace for CSS	13
Figure 6:	Loading into CSS after the workspace has been set.....	14
Figure 7:	After CSS has loaded, select the <i>Workbench</i> button.....	14
Figure 8:	Right click on the empty space in the <i>Navigator</i> pane and select Import.	15
Figure 9:	Navigating the <i>Import</i> window and finding the <i>NSFS_OPIs</i> directory with the relevant files.....	16
Figure 10:	Screenshot of the combined operating environment, which is launched through the <i>NSFS.py</i> Python program.....	18
Figure 11:	Diagram of the three components of IPOTS, where the connections between the different components are illustrated and labeled.	19
Figure 12:	Individually operating the EPICS console.....	20
Figure 13:	The initial output of EPICS after the previous sequence of commands have been invoked.....	21
Figure 14:	Individually operating the Python-based Plant Model.....	22
Figure 15:	The initial output of the Python-based Plant Model	22
Figure 16:	Opening an <i>.opi</i> file from within the CSS environment, where the user would select <i>Open With > OPI Runtime</i>	24
Figure 17:	The 'Minimize' option on a toolbar in CSS.....	25
Figure 18:	Non-disconnected sensors and components are shown in the Top-Left, with disconnected sensors and components shown in the Bottom-Right.....	26
Figure 19:	Initializing the evaluation of the temperature, pressure, and mass flow in the system on the Supervisor PCS Panel. The three buttons required to initialize this evaluation are highlighted in the figure.....	27
Figure 20:	The Operator Main Panel where the sequencer program will display information, where the three most important parts are highlighted with the black arrows.....	28
Figure 21:	The Operator Main Panel with the "Startup to Normal Operation" sequence selected. In the Actions portion of the screen, the relevant sequence of sensors and components is highlighted while the current message is put to the Messages portion of the panel.....	29

Figure 22:	The Operator and Supervisor PCS control panels, which can be used to change the state of plant components.....	30
Figure 23:	The PPS Alarm Trip Panel, where two sensors are highlighted for both the Trip Case (left, red) and the Alarm Case (right, yellow).....	32
Figure 24:	An example of a sensor's border changing color based upon an alarm or trip criteria.....	33
Figure 25:	A close-up view of the Disable PPS button on the PCS Control Panel for the Supervisor. The Disable PPS button is in a similar position on the PCS Control Panel for the Operator.	34
Figure 26:	A situation where the Plots screen has values from a previous run still stored. Right clicking on the screen and selecting "Refresh OPI" (or pressing the F5 key) will alleviate the issue.....	35
Figure 27:	Target Cooling System graphical display where the coolant line is in orange, the secondary coolant line is purple, the demineralizing line in green, and the drain tank line in blue.....	38
Figure 28:	Subcritical Assembly Cooling System graphical display where the coolant line is in orange, the secondary coolant line is purple, the demineralizing line in green, and the drain tank line in blue.	39
Figure 29:	Secondary Cooling System graphical display where the primary coolant line is in orange, the secondary coolant line is purple, the demineralizing line in green, and the drain tank line in blue.	40
Figure 30:	Accelerator Secondary Cooling System graphical display, where the primary coolant line is orange, the secondary coolant line is purple, the demineralizing line is green, and the drain line is blue.	41
Figure 31:	Operator Main Panel, where the sequence of actions that must be performed are indicated in the right section of the panel.....	42
Figure 32:	Plant Control System (PCS) Pane from the Supervisor point of view, where every individual component can be controlled and the Plant Model can be initiated.....	43
Figure 33:	Plant Control System (PCS) Panel from the Operator point of view, where some components can be controlled by the operator based on the current procedure and mode.....	44
Figure 34:	Panel containing the various plots of plant components and sensor values with respect to time.....	45
Figure 35:	Panel containing the various sensors with alarms connected to them. The components change color depending on the Alarm and Trip status of the components in the current PPS mode.	46

List of Tables

Table 1:	Elements of the Config file with a description of what how they should be set.....	17
Table 2:	Suggested CSS panels (indicated by 'X') for the Operator Mode and the Supervisor Mode.....	23

Introduction

The Neutron Source Facility Simulator (NSFS) is comprised of several parts which, when implemented together, allow the operator to simulate the operating sequences of the Kharkov Institute of Physics and Technology (KIPT) Accelerator Driven System (ADS) cooling system. Namely, these parts are the:

- **Experimental Physics and Industrial Control System (EPICS):** Industrial Plant Control System interface which simulates the presence of components.
- **Plant Model:** Pressure, temperature, and mass flow model written in Python to simulate the values in the system based upon the status of the components in EPICS.
- **Control System Studio (CSS):** Graphical User Interface to display the values in the simulated system and allow an operator to control plant components through EPICS.

Hardware Requirements

The NSFS has been developed with the following important target operating environment parameters in mind:

- Windows® 7, 64-bit operating system
- 4 GB RAM
- 3.0 GHz per core Intel® Core™ processor

These are required for several reasons. With the complexity of the implemented control system, the CSS requires the use of the 64-bit version. As such, a 64-bit operating environment is necessary, where Windows® 7 was used to facilitate the design and testing of the system. The RAM suggestion is related to the CSS environment, where opening the relevant screens consume greater than 1GB of RAM on its own. Therefore, for a stable operating environment the suggested RAM requirement is 4 GB, where having even more would be preferable such that the user is not limited in doing other actions concurrently. Finally, the listed processor speed is important for the calculation of the model provided with NSFS, which runs in a single-threaded environment on one of the processor's cores. Therefore, the speed of the processor is the limiting factor in the speed that the included model updates with respect to time.

Additionally, the parts of NSFS are designed such that they can run concurrently on the same computer as shown in Figure 1, or independently on several computers on the same subnet as shown in Figure 2. This connectivity is facilitated by the distributed nature of the EPICS, in which the industrial components are distributed throughout a facility but included on the same subnet.

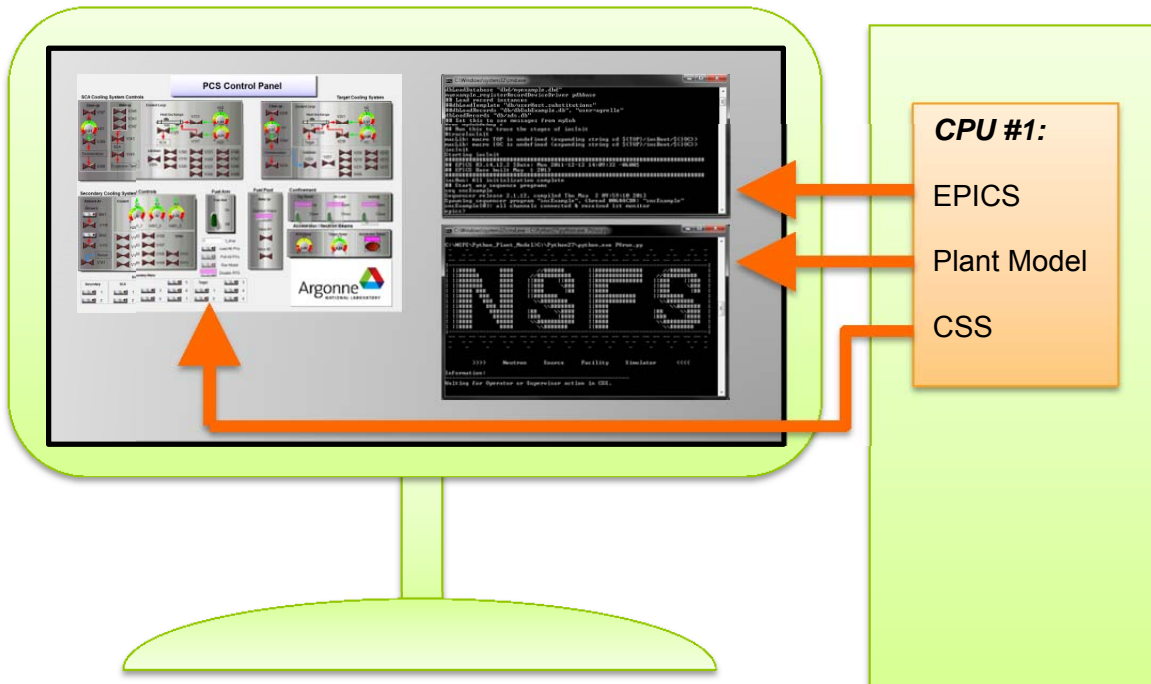


Figure 1: The option of running the entire NSFS suite on a single workstation that the operator then controls.

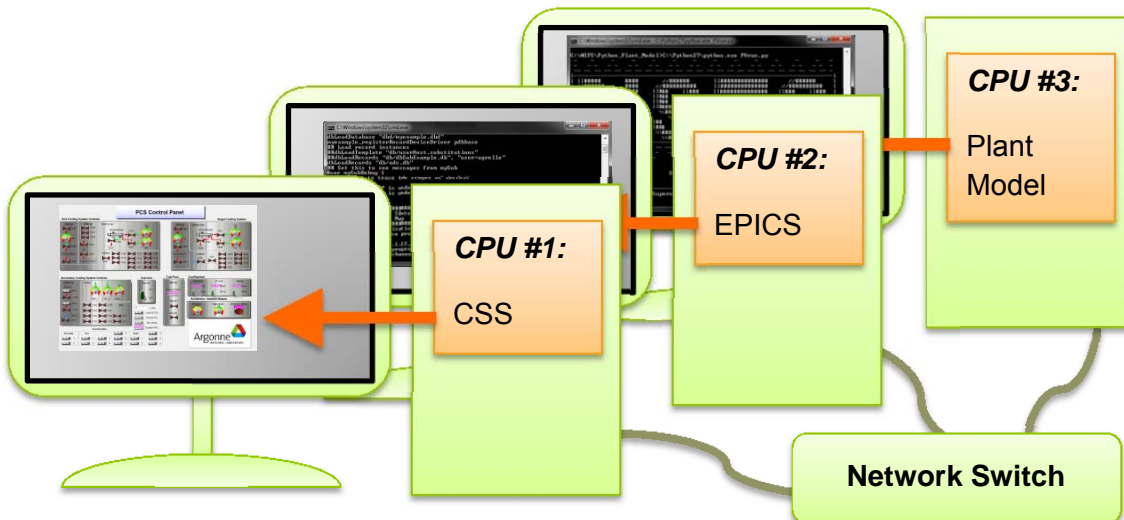


Figure 2: The option of running the different components of NSFS on separate systems, where the operator controls the simulator from CPU #1.

In such a situation, the various control elements and display screens may also be distributed across several computers under the subnet. Therefore, though NSFS can be entirely run on a single computer, it may be preferable to the user to run portions of NSFS on different computers. One example includes having the displays showing the Subcritical Assembly Cooling System, the Target Cooling System, and the Secondary Cooling System on large monitors above the user on a wall display. These displays may be operated on a different system than the computer the operator sits behind, and as such the implementation of NSFS is then split across multiple computers, as shown in Figure 3:

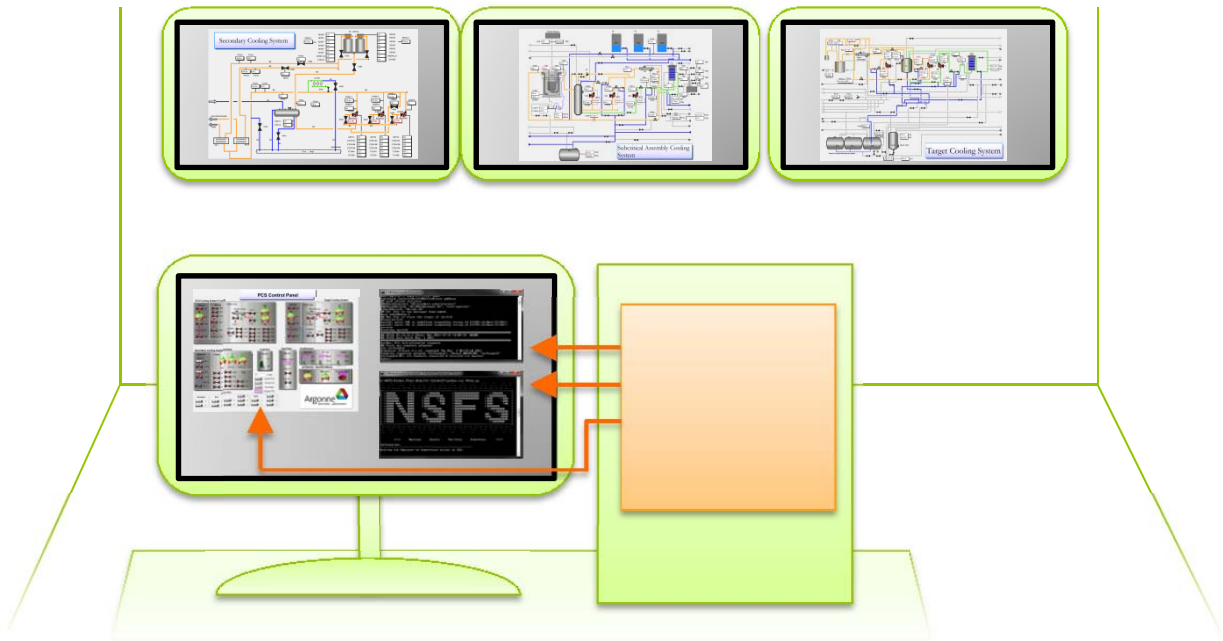


Figure 3: The option of running CSS on several different displays (potentially on different computers for each display) while controlling the system from a central workstation utilized by the operator.

A second example, if the operator's computer does not have a sufficiently fast processor as listed in the Hardware Requirements, but does have an appropriate operating environment otherwise. In this case, the operator's computer is capable of running the graphical displays but does not have the capability to run the Plant Model adequately. Such a setting could necessitate running the Plant Model Python program on a different, faster computer, while the operator controls the model from the CSS displays on another computer, as shown in Figure 4.

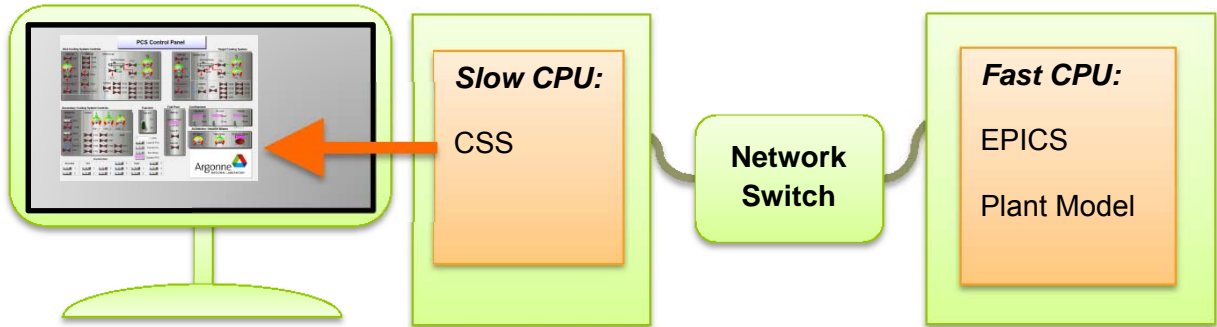


Figure 4: The option of running the Plant Model on a different, faster computer than the one that the operator utilizes.

The hardware configurations described above do not provide an exhaustive list of all available options. As such, the user is encouraged to find the solution that best fits their needs, within consideration of the prescribed system requirements for a working system. In any case, the following instructions regarding the setup of the software must be followed for every system that will be utilized as part of NSFS.

Prerequisites

The following sections describe the main NSFS components. Versions of each are located in the NSFS directory or can be obtained from the internet. Using a newer version of these components is not guaranteed to work as a newer version may remove or change features that the Plant Model depends upon.

The installation requires Administrative Privileges on the system.

Java SE Development Kit Version 7 – 64-bit

The Java development version is needed to run *Eclipse*-based software such as Control System Studio, among others. Your system may already have Java installed. If not, use the version from the NSFS directory. If run from the command line in Windows, the argument `/s` can be passed after the executable file for a silent installation.

MinGW

The EPICS installation necessitates a C compiler for itself and related programs. In the case of this Windows implementation, the MinGW compiler is suggested. As such, NSFS is only guaranteed to work if the MinGW compiler is utilized to compile EPICS for your machine.

MinGW can be obtained online, or it can also be found in the NSFS directory. There will be several steps in the installation where the 'Next' button may simply be clicked, except for the option regarding compilers. At the compiler step, make sure to choose at a minimum the 'C Compiler'.

Strawberry Perl 5.16.3.1 – 32bit

There is a Perl script that is used in the installation of EPICS on each machine. As such, it is mandatory to install a version of Perl that can execute this script appropriately. For the case of this Windows EPICS installation, the version of Strawberry Perl in the NSFS directory can be utilized. For this installation, the default settings are recommended and the installation is as simple as pushing the 'Next' button several times.

7-Zip (optional)

To install EPICS, the archive must be extracted from the .zip format found in your NSFS installation distribution. The program 7-Zip is suggested – but not required – for this step, you may use any program with similar functionality to complete this task. If you choose to install 7-Zip at this time, then it is in the NSFS directory.

Python 2.7.5 32-bit

The Plant Model developed for NSFS utilizes the Python computing language to determine the temperature, pressure, and mass flow rate throughout the system. Therefore, Python must be installed in addition to EPICS on the system that will run the Plant Model. Specifically, a version 2.7.x of Python is needed along with a few added modules; not all of the necessary modules are available in Python 3.x, so NSFS will not work successfully under the new Python 3.x paradigm. It is important to note that Python 2 and Python 3 are maintained concurrently and are not completely compatible with each other.

An installer for the 2.7.5 32-bit version of Python is included in the *Prerequisites>Plant_Model* directory within your NSFS directory. This can be utilized when installing the Plant Model, or you may download it from the internet, however NSFS is only guaranteed to work with the included version of Python. After installing Python, there are four additional modules that must be installed for NSFS to function correctly, which are listed below.

Scipy 0.12.0

To evaluate the Plant Model, some added scientific tools are used from the Scipy open-source package. For your convenience, SciPy 0.12.0 has been provided in the *Plant_Model* directory. A newer version may be downloaded from the internet, however NSFS is not guaranteed to work with the new version.

Numpy 1.7.1

The Plant Model solves a matrix representation of the system at every time step using a numerical solver included in the NumPy package. As such, this module is mandatory for the Plant Model to correctly run. The 1.7.1 version of NumPy has been included in the *Plant_Model* directory. A newer version may be downloaded from the internet, however NSFS is not guaranteed to work with the new version.

PyEpics 3.2.1

To communicate with EPICS from the Python interface, a module called PyEpics must be installed. Version 3.2.1 is included in the *Prerequisites>Plant_Model* directory. A newer version may be downloaded from the internet, however NSFS is not guaranteed to work with the new version.

Matplotlib-1.2.1.win32

Elements of the numerical solver utilize the `pylab` feature found in Matplotlib. As such, it is necessary for this package to be installed. At the Python command line, a test of this prerequisite can be taken by invoking the command `import pylab`. If this command succeeds, then this prerequisite has been accommodated.

wxPython 2.8

To help facilitate the implementation of NSFS on a single workstation, a GUI has been provided to launch EPICS, CSS, and the Plant Model using a simple interface. This interface utilizes the wxPython module to draw the GUI elements to the screen and receive feedback from the user to begin the various elements of NSFS. Therefore, this element is optional depending on how NSFS will be implemented on your specific system, but it is recommended that it is installed with the other Python modules.

Installation / Set-up

The installation of the NSFS focuses on the three parts: EPICS, installing CSS, and installing the Plant Model. The installation of each of these is important for every machine that will be a part of the NSFS training environment. The necessary files for this can be found in their respective folders under the root directory of the NSFS.

When installing the various components of NSFS, it is important that you take note of where they are being installed on the local machine. At the onset, it is helpful to identify whether you plan to install everything under the *root* directory (such as `C:\`) or another directory such as `C:\NSFS\`. This choice will be important later in the installation of EPICS, and will further be referred to as your `\root\` directory.

Installing EPICS Base

Prior to installing EPICS, the above listed Prerequisites must be addressed such that the system is ready. The files needed to install EPICS are included in the *EPICS* directory under the NSFS directory. These files should include:

- `baseR3.14.12.2.tar.gz`
- `re2c.exe`
- `seq-2.1.12.tar.gz`
- `win32.bat`

These files are necessary for the installation of EPICS. First, use 7-Zip (or your equivalent archive extractor) and extract `baseR3.14.12.2.tar.gz` to your `\root\` installation directory.

In the following documentation, some individual steps will be boxed-off from the text to help alert the reader to sub steps pertaining to a particular step in the installation process. Many steps have an *alternate option* below the box that can be performed in lieu of the steps in the box. The first such step pertains to the win32.bat file:

Editing the win32.bat file

After the extraction is finished, the win32.bat file under the \root\base-3.14.12.2\startup\ directory must be modified to match your system. The win32.bat file must contain the following lines:

- set PATH=\root\strawberry\perl\bin;%PATH%
- set PATH=\root\MinGW\bin;%PATH%
- set EPICS_HOST_ARCH=win32-x86-mingw
- set PATH=\root\base-3.14.12.2\bin\%EPICS_HOST_ARCH%;%PATH%
- set PATH=\root\seq-2.1.12\bin\win32-x86-mingw;%PATH%

Where again, the \root\ directory is your chosen installation directory such as C:\ or C:\NSFS\. The win32.bat file provided in your NSFS distribution under the EPICS folder has these lines already configured for the case when \root\ = C:\.

ALTERNATE OPTION: Copy the provided win32.bat file from the EPICS installation directory to your \root\base-3.14.12.2\startup\ directory.

Once the win32.bat file has been successfully changed or overwritten, open up the Windows shell and run the win32.bat file to configure your environment to properly compile the EPICS installation. On Windows 7, the hotkeys for this action would be [Windows Key] + c + m + d + [Enter]. Once in the shell (or 'command prompt') change your directory to \root\base-3.14.12.2\startup\ (command 'cd' in the Windows shell) and execute the win32.bat file.

After this has been executed, change directories to the EPICS directory (\root\base-3.14.12.2\) from the Windows shell. Once in this directory, type the command make to compile EPICS for your workstation. This process should take a few minutes and will have very verbose output. Assuming there are no errors in the process, move to the next step: installing the sequencer.

The portion of EPICS that controls automatic execution of sequences of action is called the *Sequencer*. This element of EPICS is referenced in the win32.bat file as \root\seq-2.1.12\, where the version number corresponds to the version of the sequencer. For the case of the NSFS distribution provided, the version should be 2.1.12.

First, extract the *Sequencer* archive using 7-Zip (or your equivalent archive extractor) to the \root\ installation directory. Once this has completed, copy the file rec2.exe to the \bin\ directory of your MinGW installation (this could be, for example, C:\MinGW\bin\). The file rec2.exe is found in the EPICS directory of your NSFS distribution.

The Sequencer RELEASE File

Next, amend the file `\root\seq-2.1.12\configure\RELEASE` such that it contains the declaration of the location of your EPICS directory:

- `EPICS_BASE=\root\base-3.14.12.2`

Where, for example, `EPICS_BASE=C:\base-3.14.12.2` would be the line if `\root\` is `C:\`.

The CONFIG_SITE File

Next, amend the file `\root\seq-2.1.12\configure\CONFIG_SITE` such that it contains the following line:

- `LEMON = \root\seq-2.1.12\bin\win32-x86-mingw\lemon.exe`

Where, for example, `LEMON = C:\seq-2.1.12\bin\win32-x86-mingw\lemon.exe` would be the line if `\root\` is `C:\`. Once this is complete, the *Sequencer* is ready to be compiled. In the Windows shell, navigate to the *Sequencer* directory and issue the `make` command. This process should take a few minutes and will have very verbose output. Assuming there are no errors in the process, move to the next step: the Perl script.

The Perl Script

In the `\root\` directory, make a directory `epics` and within that directory, make a directory called `top`. If accomplished using the Windows shell, this would be done by navigating to the `\root\` directory and issuing the following commands:

- `mkdir epics`
- `mkdir epics\top`

After this, change directory (`'cd'`) into `epics\top\` and issue the following command into the Windows shell:

- `perl \root\base-3.14.12.2\bin\win32-x86-mingw\makeBaseApp.pl -t example NSFS`

Where, the following example would be the command if `\root\` is `C:\`

- `perl C:\base-3.14.12.2\bin\win32-x86-mingw\makeBaseApp.pl -t example NSFS`

Next, run the following command in the Windows shell:

- `perl \root\base-3.14.12.2\bin\win32-x86-mingw\makeBaseApp.pl -i -t example NSFS`

Where, the following example would be the command if `\root\` is `C:\`

- `perl C:\base-3.14.12.2\bin\win32-x86-mingw\makeBaseApp.pl -i -t example NSFS`

After executing this command, the Perl script will ask “*What architecture do you want to use?*”, where you should type `win32-x86-mingw` as the answer and push *Enter*. Next, it will ask “*What application should the IOC(s) boot?*”, where you should leave it blank and push *Enter*.

At this point, the make command must be executed to create the basic file structure of the EPICS installation. Change the directory of the Windows shell to `\root\epics\top\` and issue the `make` command, where if `C:\` is `\root\` then the command would be:

- `C:\epics\top>make`

The EPICS RELEASE File

After this, the `top` directory should contain several more files and folders. These files will need to be updated or modified to include the components and sequences of NSFS. First, the file `\root\epics\top\configure\RELEASE` should be amended to contain the following line:

- `SNCSEQ=\root\seq-2.1.12`

Where, for example, `SNCSEQ=C:\seq-2.1.12` would be the line if `\root\` is `C:\`.

The next installation step focuses on the `st.cmd` command, which is important in launching EPICS.

Editing the file `st.cmd`

Next, the file `\root\epics\top\iocBoot\iocNSFS\st.cmd` should be amended to the following lines:

- `seq sncNSFS`

Where `", user=YOURUSERNAME"` can be deleted. Next, the following lines must be commented out, as shown below:

- `##dbLoadTemplate "db/userHost.substitutions"`
- `##dbLoadRecords "db/dbSubExample.db", "user=YOURUSERNAMEHost"`

After these have been commented out, add the line about the NSFS records database under the comment regarding the records, as shown below:

- `## Load record instances`
- `dbLoadRecords "db/NSFS_database.db"`

ALTERNATE OPTION: Copy the file `st.cmd` from the EPICS installation directory to your `\root\epics\top\iocBoot\iocNSFS\` directory.

The `NSFS_database.db` File

Once `st.cmd` has been setup, copy the `NSFS_database.db` file from the EPICS directory in the NSFS distribution to the `\root\epics\top\db\` directory, overwriting a previous file if necessary.

The `sncNSFS.dbd` and the `snsNSFS.stt` Files

Next, copy the `sncNSFS.stt` and the `sncNSFS.dbd` files from the EPICS directory in the NSFS distribution to the `\root\epics\NSFSApp\src\` directory, overwriting a previous file if necessary.

Editing the file `sncProgram.stt`

Next, navigate to the directory `\root\epics\top\NSFSApp\src\` to the `sncProgram.stt` file. First, edit `sncProgram.stt` to include the proper sequencer file, where the only line in the file should be:

- `#include "../sncNSFS.stt"`

ALTERNATE OPTION: Copy the file `sncProgram.stt` from the EPICS installation directory to your `\root\epics\top\NSFSApp\src\` directory.

Editing the file `Makefile` under `\root\epics\top\NSFSApp\src\`

Next, open the `Makefile` in `\root\epics\top\NSFSApp\src\` and modify it such that under the comment “# Build `sncExample` into `NSFSSupport`” it references the proper sequencer program. The two lines that must be edited are shown below:

- `NSFS_DBD += sncNSFS.dbd`
- `NSFSSupport_SRCS += sncNSFS.stt`

ALTERNATE OPTION: Copy the file `Makefile` from the EPICS installation directory to your `\root\epics\top\NSFSApp\src\` directory.

Finishing the EPICS Installation

Once this step has been completed, change the directory of the Windows shell to `\root\epics\top\` and issue the `make clean` command, as shown below if `C:\` is the `\root\` directory:

- `C:\epics\top>make clean`

This will compile your local EPICS installation to contain the components contained in the `NSFS_database.db` file, and the plant sequences, PPS values, and related procedures found in the `sncNSFS.stt` file. After this operation has finished, execute the `make` command again, as shown below:

- `C:\epics\top>make`

After this operation has finished, the installation of EPICS is complete.

Installing the Plant Model

To install the Python-based Plant Model, copy the `Plant_Model` directory from your NSFS directory to your `\root\` installation directory. Assuming all of the prerequisites have been addressed, then the Plant Model is ready to run.

Installing CSS

To install the CSS portion of NSFS, first navigate to the CSS directory in your NSFS directory to find the `CSS_EPICS_3.2.14.zip` file. Extract this directory to your chosen installation directory for NSFS, `\root\`. At this point, CSS is ready to be configured and you can navigate into the directory and execute the `css.exe` file by either double-clicking or executing it from the Windows shell.

At this point, CSS will ask you to select a workspace, as shown in Figure 5. Here, you should navigate to `\root\` and create a new directory that will function as the Workspace directory for CSS. Alternatively, the workspace directory can be placed within the `CSS_EPICS_3.1.5` directory.

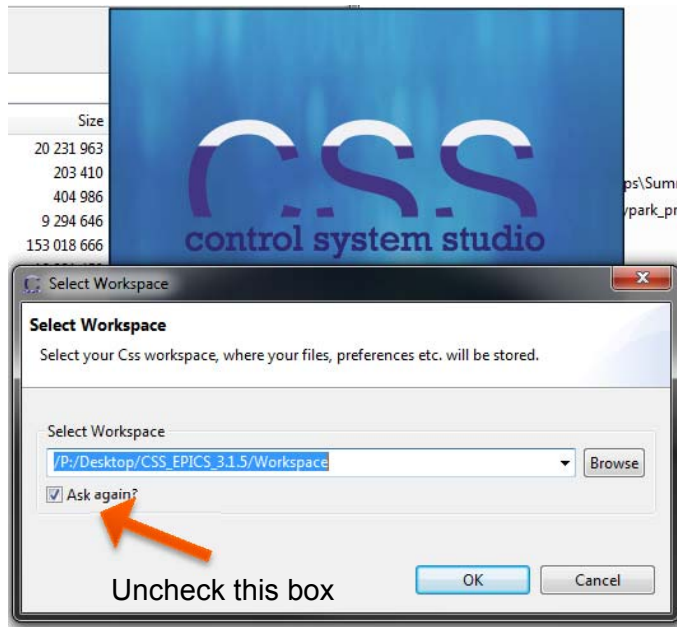


Figure 5: Initially setting the workspace for CSS

After the workspace is set, make sure to 'uncheck' the *Ask Again* box so that CSS will open automatically the next time. Once *OK* has been clicked, CSS loads into the workspace, as shown in Figure 6.

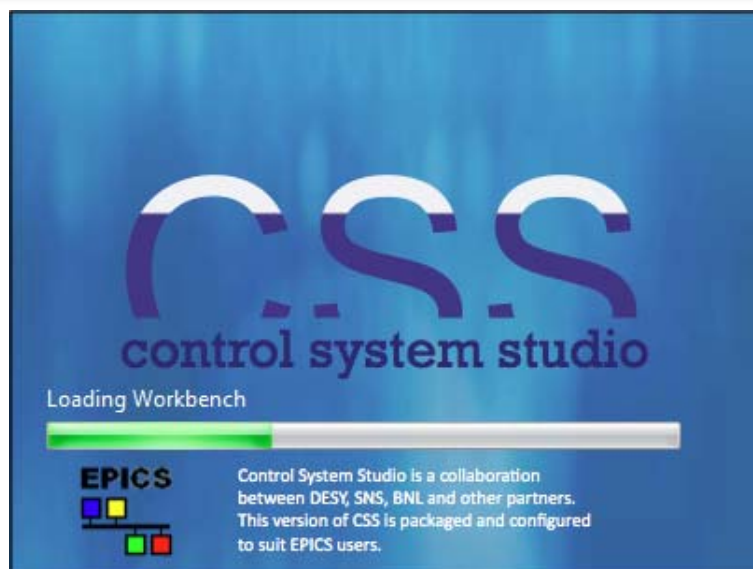


Figure 6: Loading into CSS after the workspace has been set.

Once CSS has loaded, select the *Workbench* option at the top right of the *Welcome* screen, as shown in Figure 7.

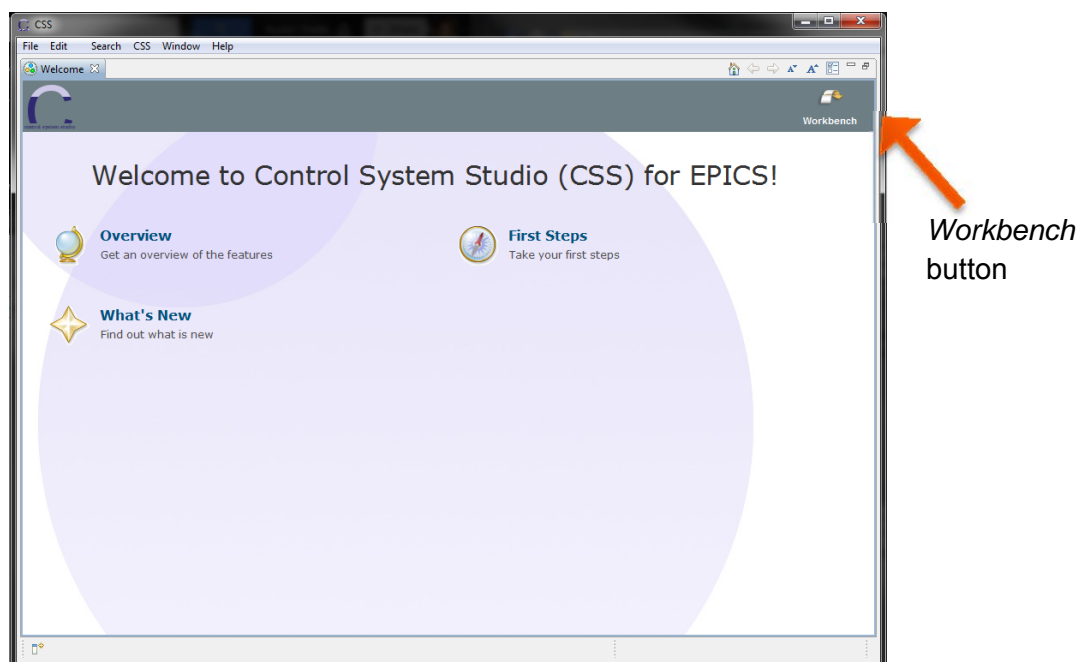


Figure 7: After CSS has loaded, select the *Workbench* button

Once in the *Workbench* mode, right click on the white space in the *Navigator* pane on the left and select the *Import* option, as shown in Figure 8.

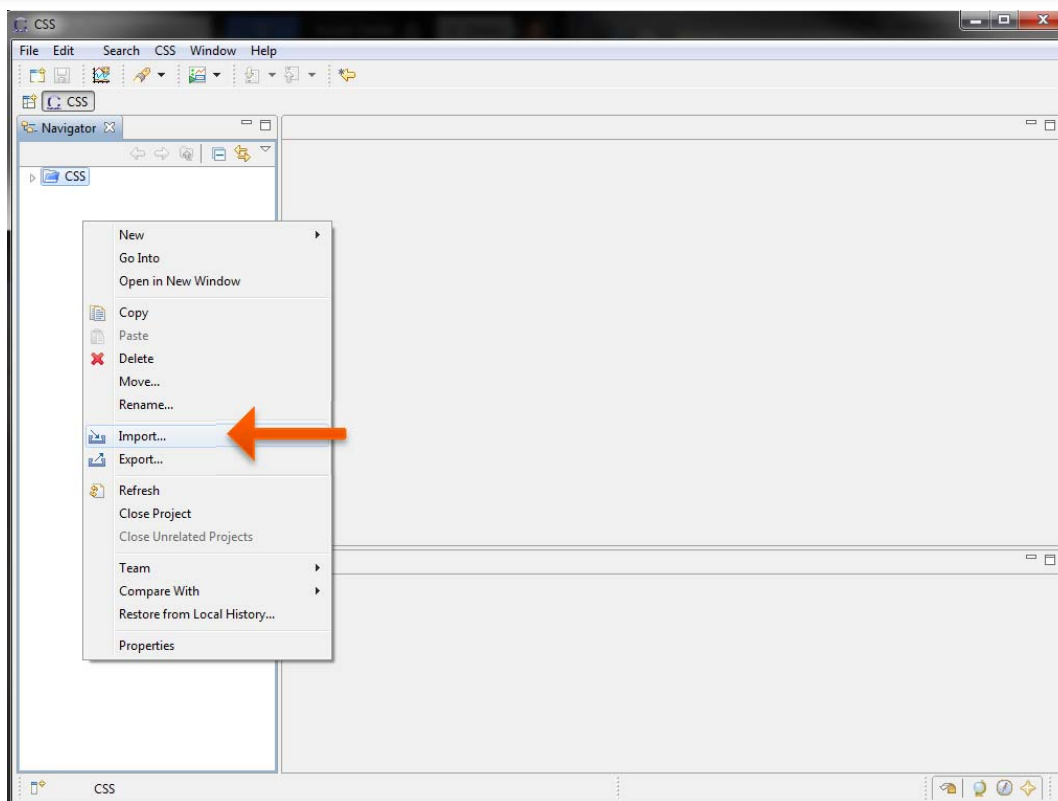


Figure 8: Right click on the empty space in the *Navigator* pane and select *Import*.

In the *Import* window, select *File System* from the *General* category. In the next screen navigate to your *CSS_EPICS_3.2.14* directory and find the *NSFS_OPIs* directory, which contains all of the relevant files to run the NSFS display screens. Select this folder and then select all of the files in it, as shown in Figure 9.

The *NSFS_OPIs* directory is where all of the plant schematic screens and control panels are located. An *.opi* file is the file where one screen and all of its elements are stored. The location of this directory will be need to be known to open the screens necessary to run NSFS.

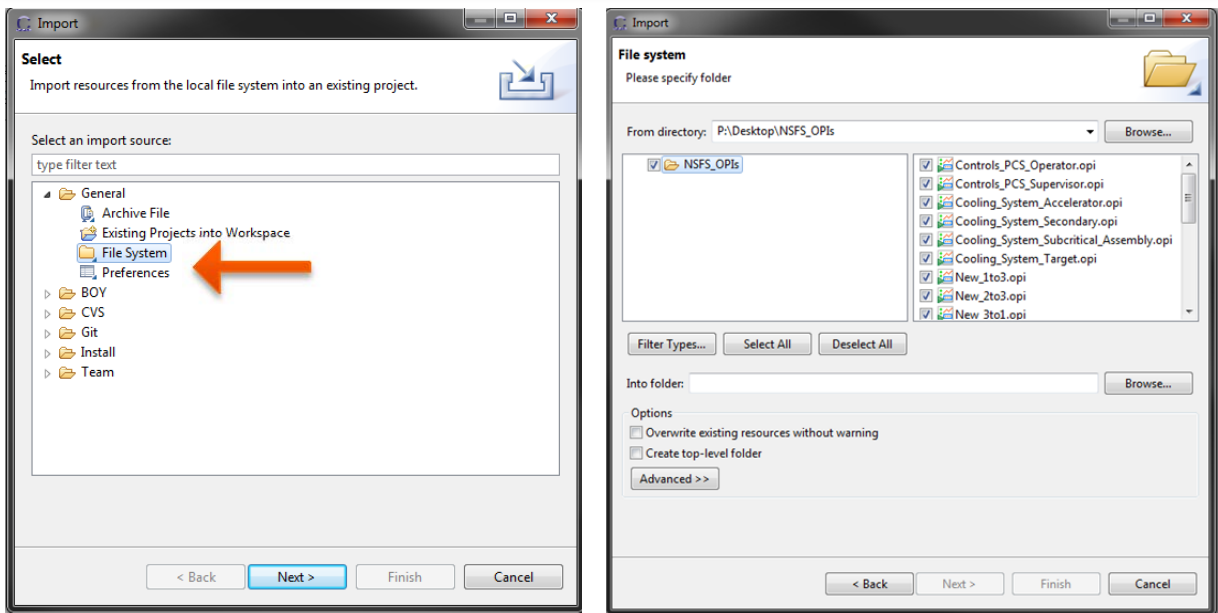


Figure 9: Navigating the *Import* window and finding the *NSFS_OPIs* directory with the relevant files

With this, CSS as it relates to NSFS is now installed and configured. The next time the `css.exe` file is executed, the program will open directly into the *Workbench* mode so that individual OPI files can be loaded. Additionally, with CSS configured, the NSFS combined operating environment (explained below) will be able to launch all of the necessary OPI files at once.

Finalizing Installation: Combined Operating Environment

The final element of NSFS to be installed is the *combined operating environment*, which is a Python program based on wxWidgets that facilitates the operation of all components of NSFS. This allows the user to run a single program that will launch everything else through simple graphical operations on the screen. The user can choose to use this tool or to run each of the three elements separately, as will be explained in the Getting Started section.

To install the combined operating environment, copy the `NSFS.py` and `Config` files and the `\Images\` folder from the NSFS directory to your `\root\` installation directory. Once copied, the `Config` file must be modified to reference the location of the NSFS component software, listed with their description in Table 1.

Table 1: Elements of the `Config` file with a description of what how they should be set

Config file element	Description of what should be referenced
<code>CSS_Workspace_dir</code>	Where the <i>NSFS_OPIs</i> directory is located
<code>CSS_dir</code>	Where <i>CSS_EPICS_3.1.5</i> or your CSS version is located
<code>Python_dir</code>	The location where your Python 2.7.x is installed
<code>Model_dir</code>	Where the <i>Plant_Model</i> directory is located
<code>eqics_seq_dir</code>	The location of your <code>\epics\top\</code> directory
<code>epics_base_dir</code>	Where EPICS is installed, such as <code>C:\base-3.14.12.2</code>
<code>epics_app_name</code>	The app name, which should be left as <i>NSFS</i>

In the `Config` file, the delimiter between the file element on the left and its value on the right is the equals sign (=). The characters on the left of the equals sign should be left alone, as a change in spelling will prompt NSFS to incorrectly read the file. Additionally, the spelling on the right side of the equals sign should be accurate as the NSFS program will be unable to handle an incorrectly linked directory. When adding a directory on the right side, do not add an additional backslash. White-space () on either side of any element is ignored, however white-space in the middle of an element will cause an error, i.e. (`= C:\Directory` ' is the same as `= C:\Directory`', but `= C:\ Directory\` is not the same as `= C:\Directory`.)

If any one of these directories are listed improperly, NSFS should give a warning to the console regarding the component that was not able to start. In the case that an error is given, the directories that are associated with this component should be checked to make sure that they are installed correctly and listed in the `Config` file correctly.

Getting Started

With the installation of all the Prerequisites and the Installation / Set-up complete, NSFS is ready for operation. The desired use of NSFS could involve a single computer hosting all of the operating components, or these components may be distributed across several computers. In either case, the following tools and instructions will be useful for operating the different components of NSFS.

Quick Start Guide

The easiest way to run NSFS where the environment is properly configured for *EPICS*, *CSS*, and the *Python Model* is to run the *Combined Operating Environment*. This program utilizes the wxWidgets framework for creating GUI elements using Python. As such, the program is a Python program called *NSFS.py*, which will access the various components of NSFS through the directories listed in the *Config* file in the same directory. By launching the *NSFS.py* file, the user will be presented with a window that will give options to open selected OPIs or to launch simultaneously the Python Plant Model and EPICS from a preconfigured directory, as shown in Figure 10

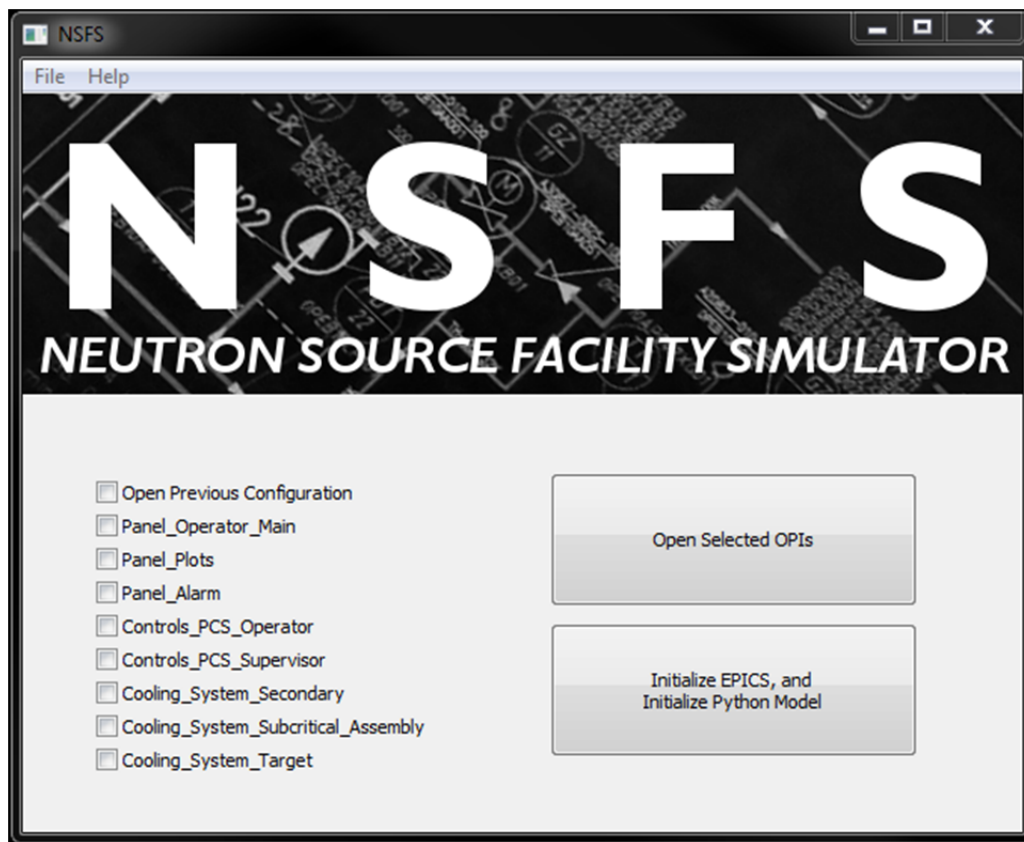


Figure 10: Screenshot of the combined operating environment, which is launched through the *NSFS.py* Python program.

By selecting the OPI files and clicking the “*Open Selected OPIs*” button, NSFS will open the selected OPI files so that the user can monitor the system on their local screen(s). This necessitates that the other two components of the system are currently active elsewhere on the subnet, namely EPICS and the Python Model. If the user instead engages the “*Initialize EPICS, and Initialize Python Model*” button, then these two components of NSFS are enacted at once. Therefore, the *Supervisor* will wish to activate the EPICS and Python Model button prior to any use of the OPI file for the *Operator* mode for operation of NSFS.

Individual Operation of NSFS Component Software

Instead of running NSFS completely on one system, the constituent components can be spread across several properly configured computers on the same subnet. If only running a certain portion of the NSFS, it would be foreseeable that the user might configure that system to only have that component installed. Still, it is advised that all of the prerequisites and the full installation are finished on every machine that will be involved with the operation of NSFS. Once this is complete, follow the instructions below for individual operation of the three NSFS components.

Figure 11 shows the general organization of the NSFS components, where the connections between the different components are illustrated and labeled. This visualization highlights important considerations of the model, for example, that a currently running EPICS environment is integral to the initialization of the Plant Model.

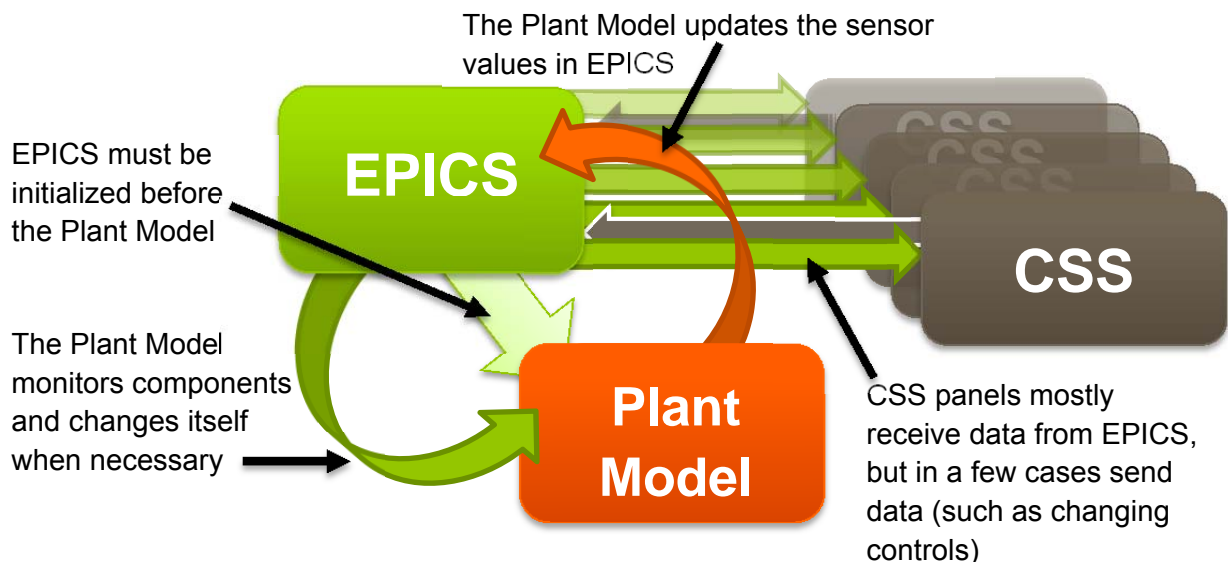


Figure 11: Diagram of the three components of IPOTS, where the connections between the different components are illustrated and labeled.

Another important observation from Figure 11 is that the CSS component of NSFS can in reality exist in several different instances, where the EPICS and Plant Model components are singular in their implementation. Therefore, the CSS component may be spread across

multiple different computers and screens, while the EPICS console and the Plant Model console will only be run on one computer or one computer each, without duplication.

Running EPICS

EPICS can be operated independently from the other components of NSFS if the user's setup requires it. All input and output associated with EPICS is done through the console (the Windows shell, `cmd.exe`, on a Windows environment). To open the shell on the recommended system, push the following key combination, in the following order:

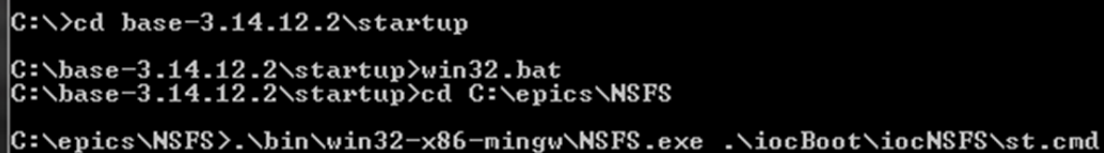
- [Windows Key] + [c] + [m] + [d] + [Enter]

Once the Windows shell is open, navigate to the directory where the EPICS base was installed, and then its subdirectory called `\startup\`. Next, invoke the file `win32.bat` in this directory to set the system's local `%PATH%` variable to contain the proper directories for running EPICS.

After invoking the `win32.bat` file under `\base-3.14.12.2\startup\`, navigate to the directory of your `\epics\top\` directory. Here, invoke the following command:

- `.\bin\win32-x86-mingw\NSFS.exe .\iocBoot\iocNSFS\st.cmd`

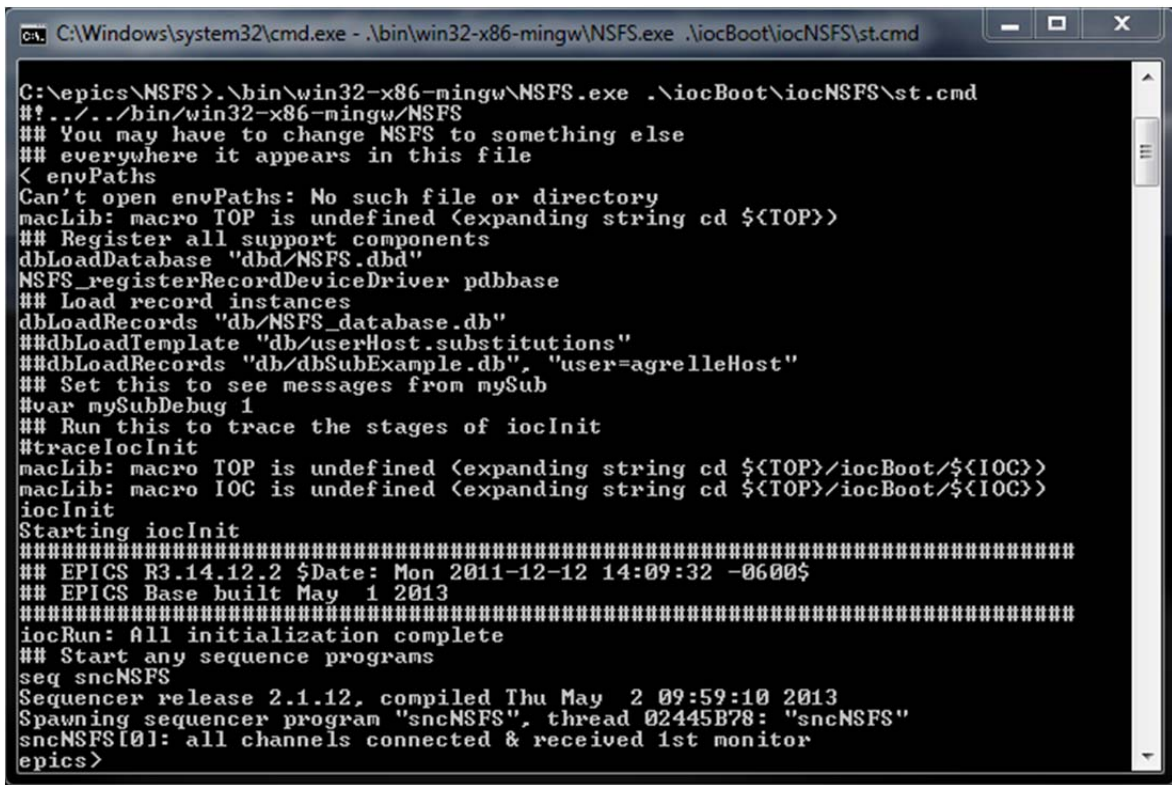
Here, it is important to note that you must be currently in the `\epics\top\` directory and that `win32.bat` must have been ran prior to this command. An example of this sequence of commands is shown in Figure 12.



```
C:\>cd base-3.14.12.2\startup
C:\base-3.14.12.2\startup>win32.bat
C:\base-3.14.12.2\startup>cd C:\epics\NSFS
C:\epics\NSFS>.\bin\win32-x86-mingw\NSFS.exe .\iocBoot\iocNSFS\st.cmd
```

Figure 12: Individually operating the EPICS console

After this command has been invoked, the user's console should show text indicating certain elements of EPICS have been loaded followed by the EPICS command line at the bottom, ready to receive text, as shown in Figure 13.



```
C:\Windows\system32\cmd.exe - .\bin\win32-x86-mingw\NSFS.exe .\iocBoot\iocNSFS\st.cmd

C:\epics\NSFS>.\bin\win32-x86-mingw\NSFS.exe .\iocBoot\iocNSFS\st.cmd
##!../bin/win32-x86-mingw/NSFS
## You may have to change NSFS to something else
## everywhere it appears in this file
< envPaths
Can't open envPaths: No such file or directory
macLib: macro TOP is undefined (expanding string cd ${TOP})
## Register all support components
dbLoadDatabase "dbd/NSFS.dbd"
NSFS_registerRecordDeviceDriver pdbname
## Load record instances
dbLoadRecords "db/NSFS_database.db"
##dbLoadTemplate "db/userHost.substitutions"
##dbLoadRecords "db/dbSubExample.db", "user=agrelleHost"
## Set this to see messages from mySub
#var mySubDebug 1
## Run this to trace the stages of iocInit
#tracelocInit
macLib: macro TOP is undefined (expanding string cd ${TOP}/iocBoot/${IOC})
macLib: macro IOC is undefined (expanding string cd ${TOP}/iocBoot/${IOC})
iocInit
Starting iocInit
#####
## EPICS R3.14.12.2 $Date: Mon 2011-12-12 14:09:32 -0600$
## EPICS Base built May 1 2013
#####
iocRun: All initialization complete
## Start any sequence programs
seq sncNSFS
Sequencer release 2.1.12, compiled Thu May 2 09:59:10 2013
Spawning sequencer program "sncNSFS", thread 02445B78: "sncNSFS"
sncNSFS[0]: all channels connected & received 1st monitor
epics>
```

Figure 13: The initial output of EPICS after the previous sequence of commands have been invoked.

This completes the initialization of EPICS. At this point, EPICS is ready to send and to receive signals to any other properly configured machines on the subnet. In the case where the user desires to run EPICS and nothing else on this computer, they can move on. In the more common case where the user also desires to run the Plant Model on the same system, the user can now begin initializing the Plant Model.

Running the Plant Model

The Python-based Plant Model can be run from any environment that the user is comfortable running a Python program on. In a properly configured system, the user simply needs to “double-click” on the Python file in the appropriate directory for the system to run the program. However, for this case, running the Python-based Plant Model through the Windows shell will be demonstrated.

Prior to running the Plant Model, EPICS must be running somewhere on the subnet, such as potentially the same computer. If the Plant Model is initialized prior to EPICS, then the Plant Model will need to be shut down and restarted once EPICS is initialized.

First, open the Windows shell (cmd.exe), and navigate to the directory that the Plant Model is in. Once inside this directory, initialize the Python 2.7.x environment and pass the file PVRun.py as an argument, as shown in Figure 14.

```
C:\>cd NSFS\Python_Plant_Model
C:\NSFS\Python_Plant_Model>C:\Python27\python.exe PVRun.py
```

Figure 14: Individually operating the Python-based Plant Model

After invoking this command, then the Plant Model is initialized and the initial output is printed to this console window, as shown in Figure 15.

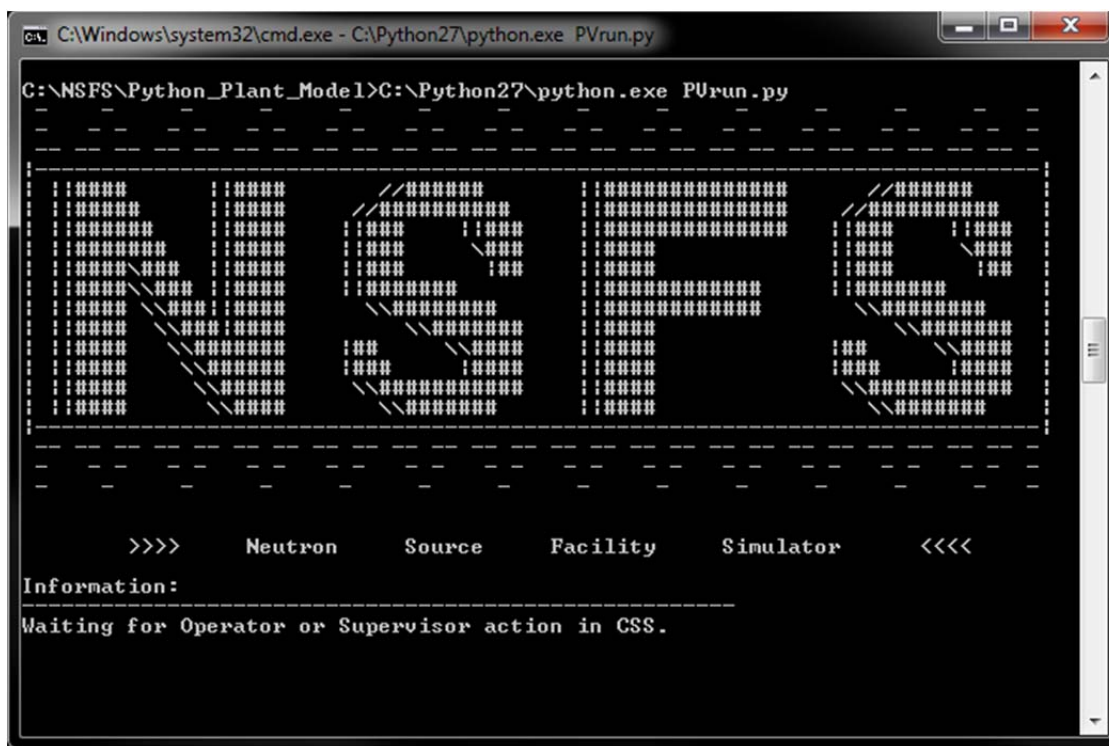


Figure 15: The initial output of the Python-based Plant Model

This completes the initialization of the Plant Model. At this point, the user should have both EPICS and the Plant Model running either on the same computer or on two computers on the same subnet. The Plant Model now waits for commands sent from the Supervisor or Operator Control Panels. The Control Panels send a signal to EPICS, and then the Plant Model monitors these signals in EPICS. When the appropriate controls are enacted on the process control panel, the Plant Model then initializes the coolant loops and begins to model the pressure, mass flow, and temperature in the three loops. The values of the sensors at each time step are posted to EPICS, which allows CSS to read the values and post them to the screen. Therefore, the three components work in unison after this point through communication facilitated by EPICS.

Running CSS

The final element of NSFS is the Control System Studio (CSS), which creates the visualizations that include plant control elements, plant diagnostic tools such as sensors and plots, and plant schematics with sensors and components. These screens are the visual representation of NSFS to the end-user, as they are used to visualize the sensors and components contained within EPICS as well as controlling the components that affect the Plant Model. As such, it is important that CSS be properly configured such that the user experience is of a high quality.

Every CSS panel is stored in a file with the extension `.OP1`. These files are self-contained and for most cases, editing one panel's file will not affect the others¹.

Additionally, each panel can therefore be opened independently of one another, allowing a scenario where each panel is potentially opened on a separate computer such as in Figure 3. This would be beneficial if the user desired to put each panel on a separate monitor where each coolant loop was displayed on a large panel.

When using the *Combined Operating Environment*, the user is given a list of available panels to open based upon their role. Many of the panels are the same in both cases, except for a few which are focused solely on actions that should be performed by the Operator or the Supervisor. As such, Table 2 lists the recommended panels for the two different modes.

Table 2: Suggested CSS panels (indicated by 'X') for the Operator Mode and the Supervisor Mode

Panel Name	Operator Mode	Supervisor Mode
Panel_Operator_Main	X	
Panel_Plots	X	Optional
Panel_Alarm	X	X
Controls_PCS_Operator	X	
Controls_PCS_Supervisor		X
Cooling_System_Secondary	X	Optional
Cooling_System_Subcritical_Assembly	X	Optional
Cooling_System_Target	X	Optional

The Operator Mode requires all panels except for the Plant Control System designed for the Supervisor, as the Operator's role necessitates the ability to view all elements of the simulation. The Supervisor could view these system elements if desired, but must mandatorily run the Plant Control System Supervisor panel, as this panel have controls that are important for running the simulation.

¹ The exception would be a panel which loads other panels as part of its normal operation. An example of this type of panel would be the Main Operator Panel, which loads different, smaller panels in the right-hand side depending on the current sequence of events.

Operator Mode

The Operator Mode in the *NSFS Combined Operating Environment* focuses on the deployment of CSS panels on the current display. As such, if the user wishes to load CSS panels on multiple computers, the Operator Mode would be the choice to be made in the *Combined Operating Environment*. However, if the user needs to open a file or files individually, then they must use the CSS interface as described below.

Using the CSS Interface

Opening a File

If the user were to open the files individually using the CSS interface, they should navigate to the .opi file, “right-click” on it, and select the Open With > OPI Runtime option, as shown in Figure 16. This will allow the .opi file to open in the mode that communicates with EPICS to send control interactions and to receive sensor and other information from EPICS.

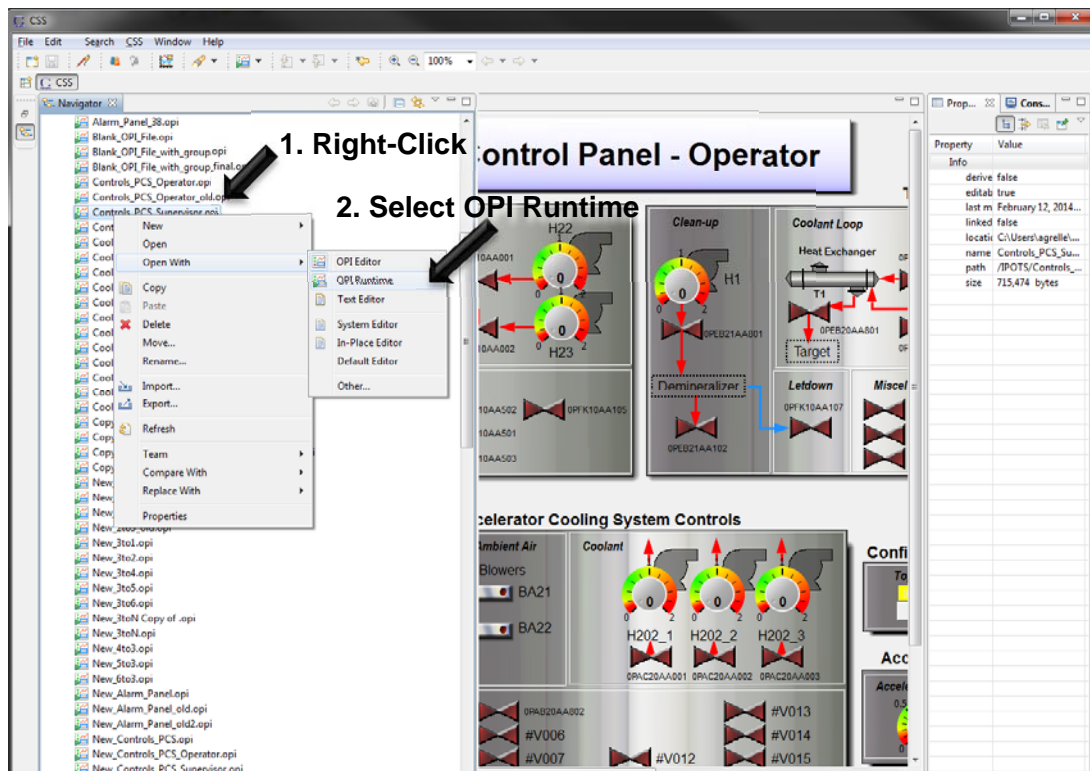


Figure 16: Opening an .opi file from within the CSS environment, where the user would select Open With > OPI Runtime.

OPI Runtime Mode Considerations

Maximizing the Display:

Once the panel is open, the user may find it useful to maximize the display, which can be achieved with the shortcut key [F8]. This will hide panels that are not useful for an end-user of CSS such as the users of NSFS.

Opening a New Window:

If the user wishes to open a new window, they can select the menu option `Window > New Window` and then drag the newly opened OPI file to this window on another monitor. This is how a user with multiple monitors will fill each monitor with individual or several different panels.

Minimizing Toolbars on the Screen:

When setting up the displays, there may be 'toolbars' that cover some of the screen such as the Properties or Navigator toolbars. To hide these, click the 'Minimize' button in the upper-right portion of the toolbar, as shown in Figure 17.

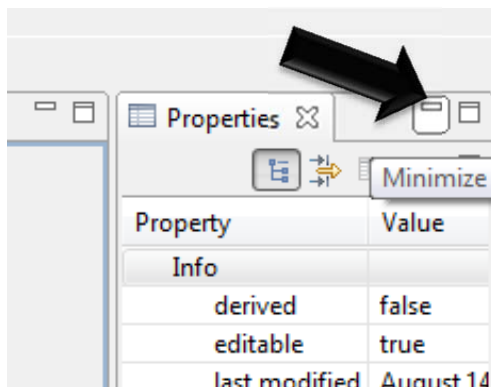


Figure 17: The 'Minimize' option on a toolbar in CSS

Disconnected Components or Sensors:

When an OPI file is opened in the *Runtime* mode the sensors and components in the file attempt to connect to the EPICS database. If there is not a component or sensor in the database by the given name, or if EPICS is not running, then the graphic will show a purple 'Disconnected' notification, as shown in Figure 18. Additionally, this could mean that CSS has yet to fully connect to the EPICS database. In this case, *Refreshing* the screen could expedite the process and is accomplished through the shortcut key [F5].

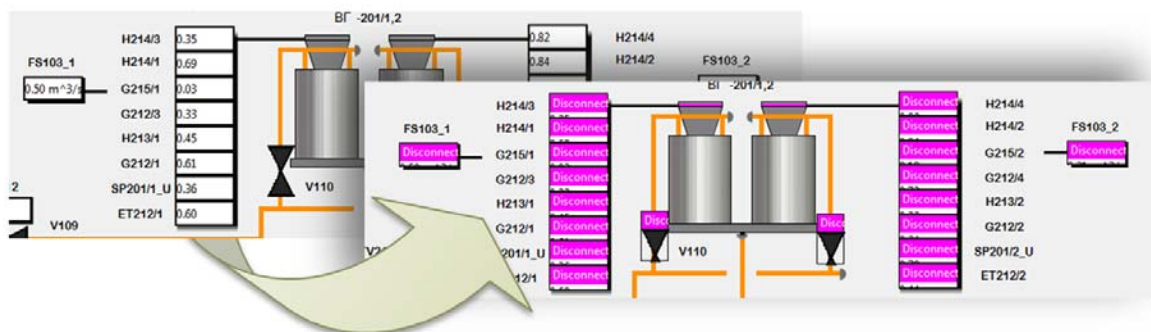


Figure 18: Non-disconnected sensors and components are shown in the Top-Left, with disconnected sensors and components shown in the Bottom-Right.

Units Displayed in Sensors:

The units that are displayed in the sensors throughout the CSS panels used in NSFS are derived from the EPICS database and are not defined in CSS itself. Therefore, these labels are a part of the EPICS distribution and the associated database file would have to be addressed if any changes were necessary. Additionally, some sensors require a conversion between the Plant Model and the display, which is handled by the Python program `PVrun.py`. Therefore, it is important to take this conversion between the Plant Model, the listed units in the EPICS database, and their eventual display on the sensors in the CSS panel into consideration when attempting to adjust sensor values.

Changing the default logo on the Operator Main Panel:

In the installation directory, the file `\root\CSS\NSFS_OPIs\Pictures\NSFS.png` can be replaced with another image of the same format and exact same name. Whichever image is located under this exact filename will be used in the Operator Main Panel.

Starting the Simulation:

One of the most important steps in NSFS is initializing the part of the Plant Model that simulates the pressure, mass flow, and temperature within the system. Once the Plant Model is running, it is constantly checking to see if the user identifying as the Supervisor is ready to begin the simulation. This is accomplished through the explicit action of clicking buttons in the Supervisor PCS Panel, as shown in Figure 19.

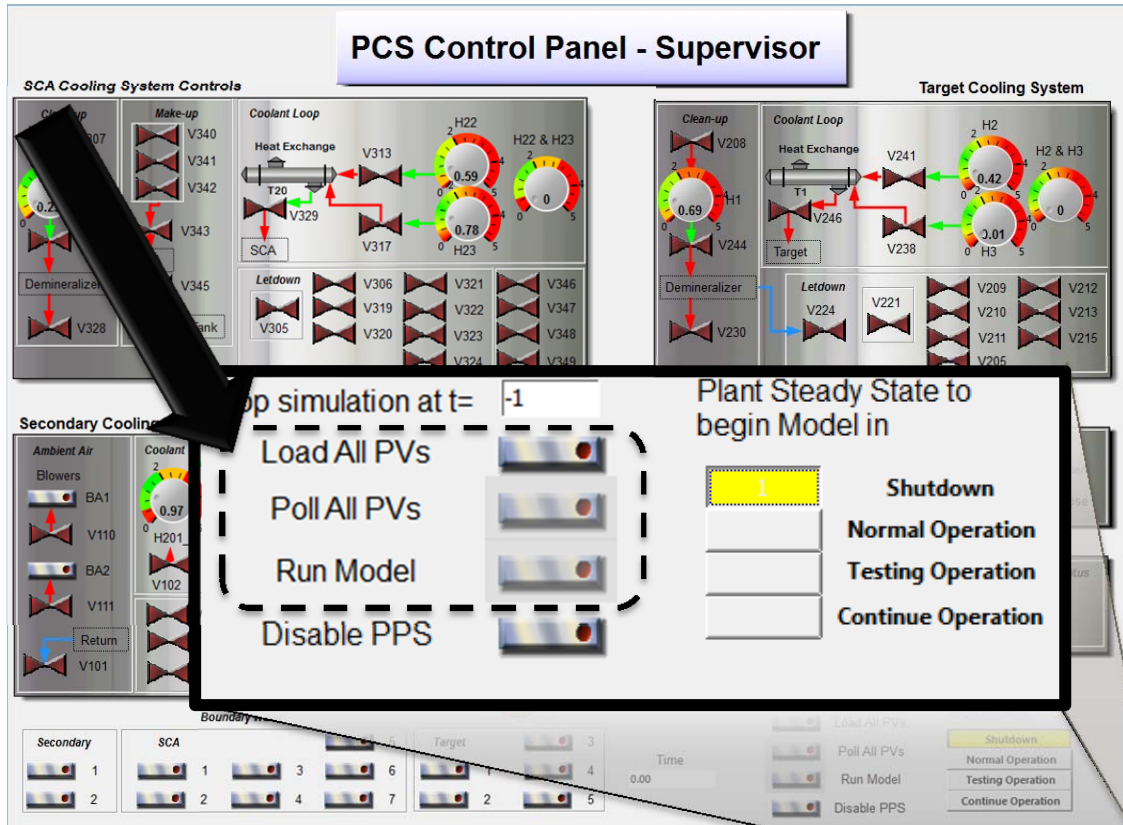


Figure 19: Initializing the evaluation of the temperature, pressure, and mass flow in the system on the Supervisor PCS Panel. The three buttons required to initialize this evaluation are highlighted in the figure.

The Supervisor must first tell the Plant Model to “Load All PVs” by clicking the Load All PVs button. This will load all sensors and components from the system in the configuration selected from the *Plant Steady State to begin Model in* section of the panel. For the case shown above in Figure 19, this will cause the Plant Model to set all of the components to the ‘Shutdown’ state. At this point, some components are ‘off’, while others are ‘on.’

The second button necessary to initialize the model evaluation is labeled as “Poll All PVs”, which will tell the Plant Model to construct the input to the model evaluation and initialize the coolant loops, if not already done. The “Run Model” button will tell the Plant Model to perform the model evaluation for as long as declared in the field labeled “t_stop,” where the input is in seconds. If the input is -1, then the Plant Model will run continuously until this value is changed.

Operating the Simulation Sequence:

With the simulation started using the instructions above, the components of the plant can be changed through either individual operator action or through the use of the sequencer program. The sequencer program will allow the operator to perform pre-defined tasks, such as starting up the system in the correct order and under the appropriate circumstances. At each time step, the Plant Model will update based upon the configuration of the components and post the resulting simulated physical sensor values to EPICS. These values will then allow the sequencer to proceed to the next step if the proper conditions are met.

The sequencer was designed from the perspective of the plant Operator, and is found in the Operator Main Panel, shown in Figure 20 with the various parts of the panel highlighted:

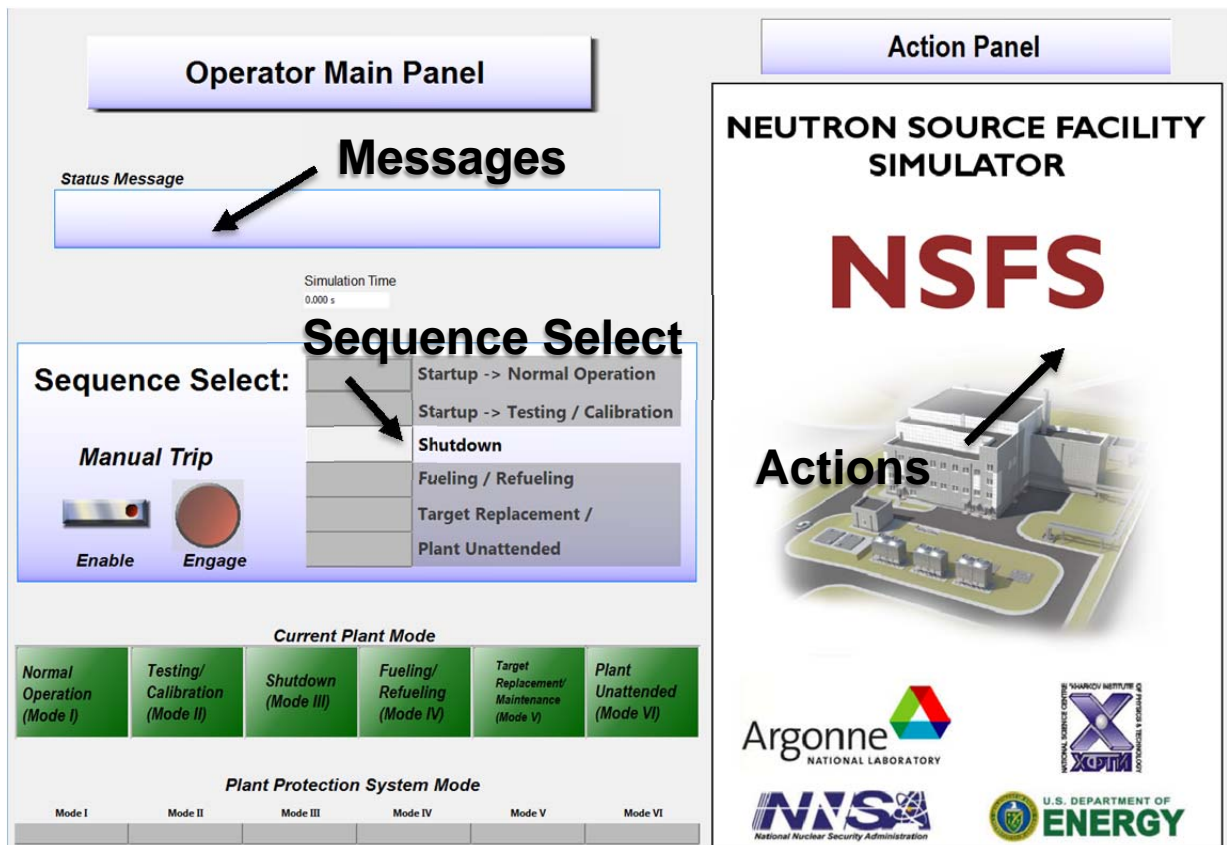


Figure 20: The Operator Main Panel where the sequencer program will display information, where the three most important parts are highlighted with the black arrows.

Once the operator is ready to begin a sequence, the eligible sequence shall be chosen from the Sequence Select portion of the panel. By clicking the Sequence Select button, the section of the screen labeled Actions will be filled with the sequence of events that will take place. In Figure 21, the 'Startup to Normal Operation' sequence has been selected with the model having begun in the 'Shutdown' mode.

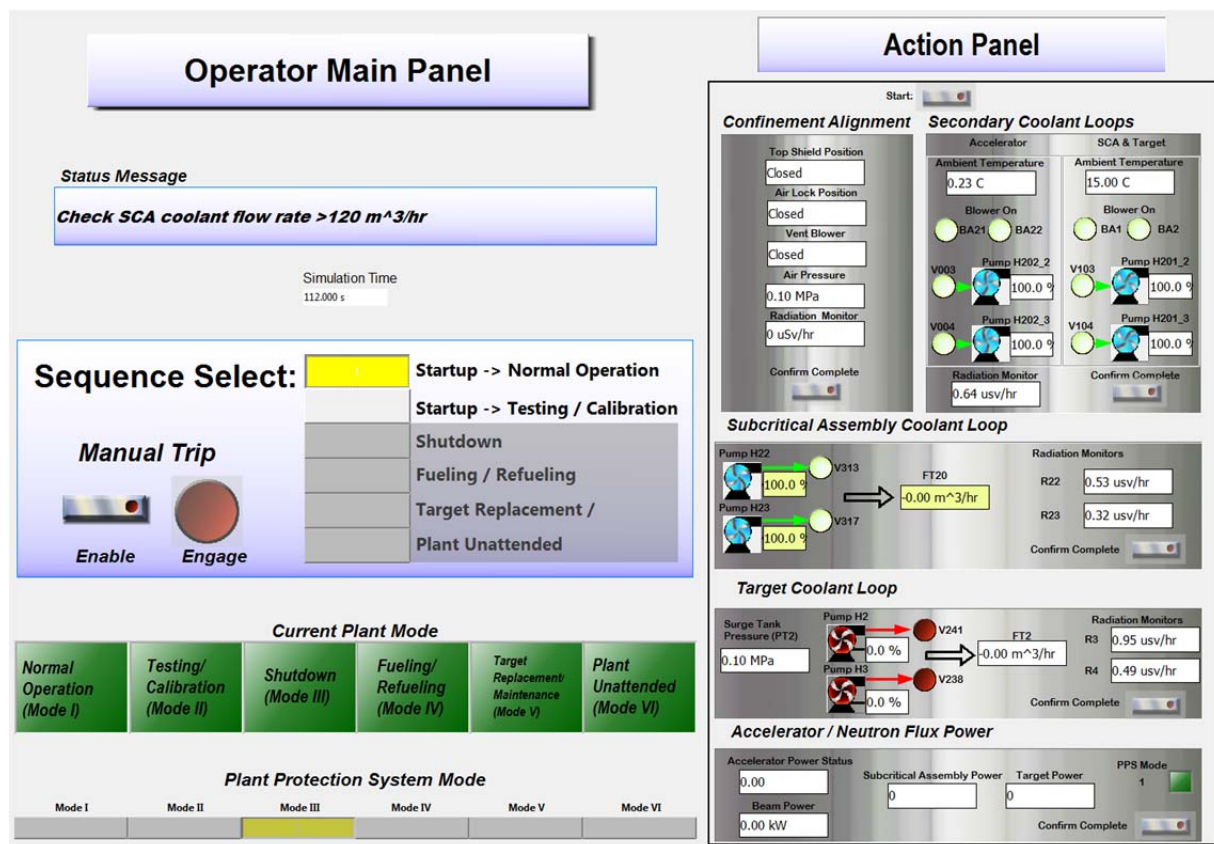


Figure 21: The Operator Main Panel with the “Startup to Normal Operation” sequence selected. In the Actions portion of the screen, the relevant sequence of sensors and components is highlighted while the current message is put to the Messages portion of the panel.

The operator then must step through the various portions of the sequence, periodically confirming that the sub-set of the sequence has been completed. Additionally, at certain points in the sequence the operator must change the Plant Protection System Mode to the new mode when appropriate.

Once a sequence has completed, the Actions portion of the Operator Main Panel will be reset to the logo, as shown in Figure 20. With the sequence complete, the operator can then progress into the next eligible sequence.

Operating Individual Components Outside of a Sequence:

The Plant Model is controlled by the sequencer program through the pre-defined operator sequences that are featured in the Operator Main Panel. Throughout the use of the simulator, the Operator and Supervisor may desire to change individual components and observe the effect they have on the Plant. In this case, the Operator and Supervisor PCS control panels, as shown in Figure 22, allow the user to change individual components which could be useful to observe less-common transients throughout the system.

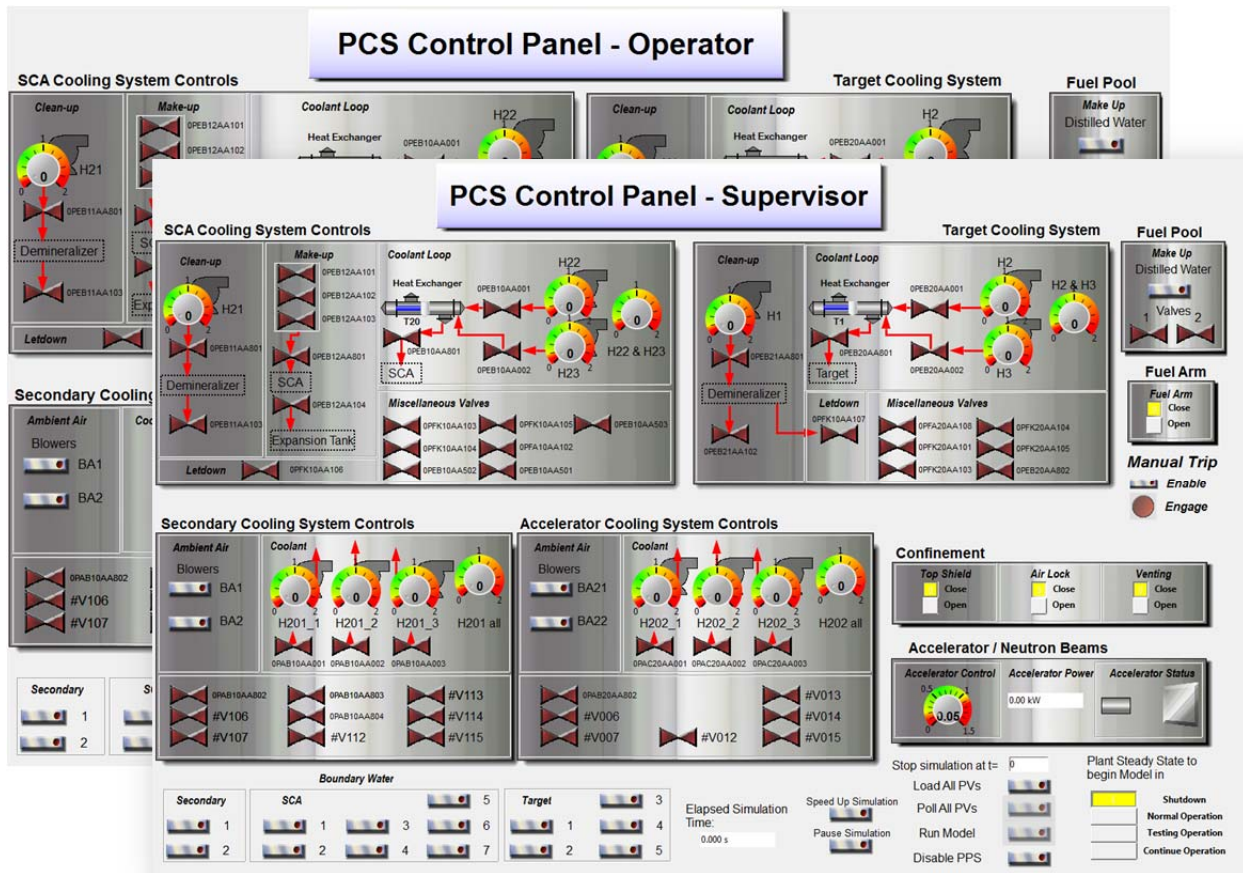


Figure 22: The Operator and Supervisor PCS control panels, which can be used to change the state of plant components.

There are differences, however, between the Operator PCS Control Panel and the Supervisor PCS Control Panel. While the Supervisor's control panel has access to all components at all times, the Operator's control panel removes access to certain elements when the Plant Protection System dictates that these controls cannot be changed.

Additionally, the Supervisor's PCS Control Panel has additional controls that are not found on the Operator's PCS Control Panel. These additional controls and features are:

- Immunity from the access restriction imposed by the Plant Protection System on certain components,
- Plant Steady State option to initialize the plant in various states,
- A button to enable the access to the trip button, and the trip button,
- A combined pump button which will control all coolant pumps in a specific loop with one single control.
- A sliding control on the heat exchangers that changes the coefficient linked with each heat exchanger, where '1' is 'normal.' The range is [0,2], where '2' is twice as high a coefficient as 'normal.'

These additional features allow the Supervisor to customize the experience for the Operator in ways that are outside of the context of the sequencer and resulting Plant Protection System. Additionally, the operator may choose to use the 'Supervisor' panel themselves so that they have full access to the myriad of components that control the Plant Model. In either case, the PCS Control Panel allows the user to have a more diverse experience using NSFS.

Observing Alarms:

The Alarm system incorporated in NSFS allows the user to experience transients that are outside of the bounds of normal operation. Though potentially disastrous in the real-world, the values that would result in an alarm can be witnessed through the simulation without harm. The values that dictate the point at which an alarm sounds are programmed into the sequencer program, `sncNSFS.stt`, which contains the actions that lead to a shutdown event if the conditions reach a point for a trip event. Therefore, it is important to address an alarm prior to the point that a trip event would be reached.

Alarms can be noticed in two separate places within the CSS panels. The first place is directly on the Alarm Panel itself, as shown in Figure 23. This panel shows the various sensor values and highlights the value in yellow if the value is in alarm and in red if it is in trip. Next to the sensor value are two buttons that can be utilized to reset the alarm signal if the operator has addressed the problem.

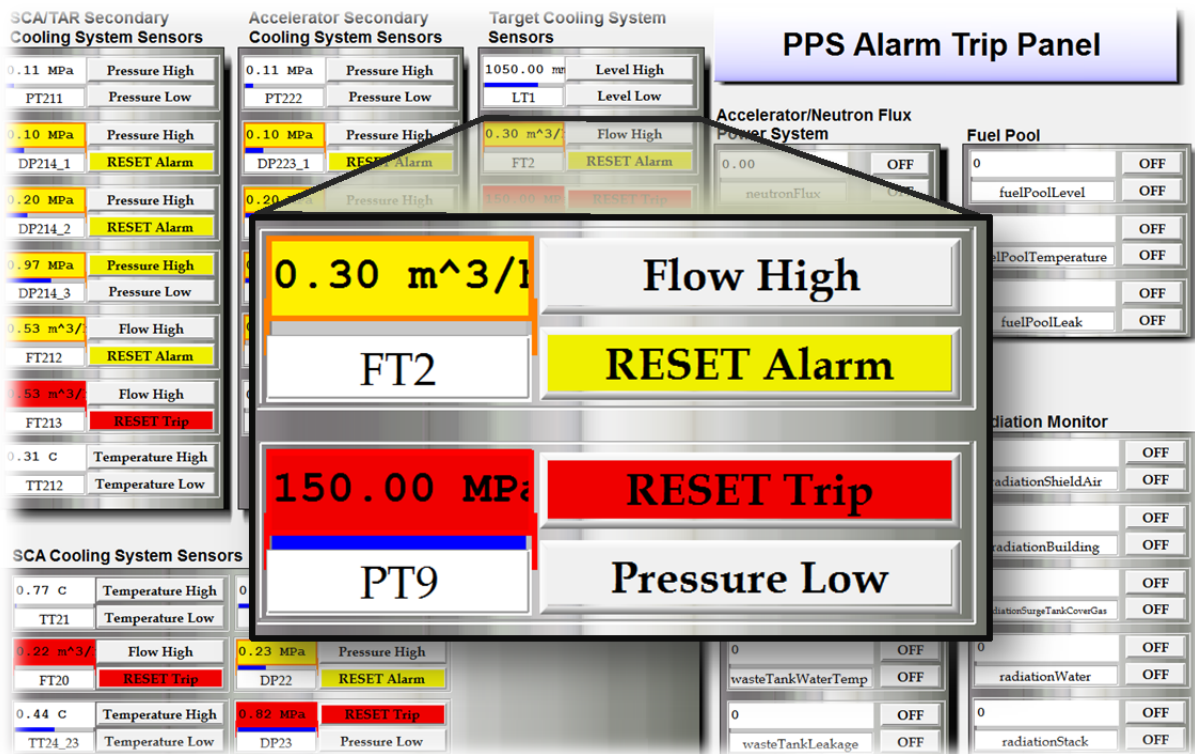


Figure 23: The PPS Alarm Trip Panel, where two sensors are highlighted for both the Trip Case (left, red) and the Alarm Case (right, yellow).

If the user clicks the *RESET* button associated with the high or low alarm or trip and the value is not within an appropriate threshold, then the button will re-illuminate until it is pressed at an appropriate time.

The second location that the user may witness an alarm or trip event is on the panels displaying the loops. Each sensor that is located on the three coolant loops will change the outline surrounding the sensor if there is a condition that is outside of the bounds of normal operation, as illustrated in Figure 24. Use in conjunction with the PPS Alarm panel, the user can find and address these alarm and trip situations quickly.

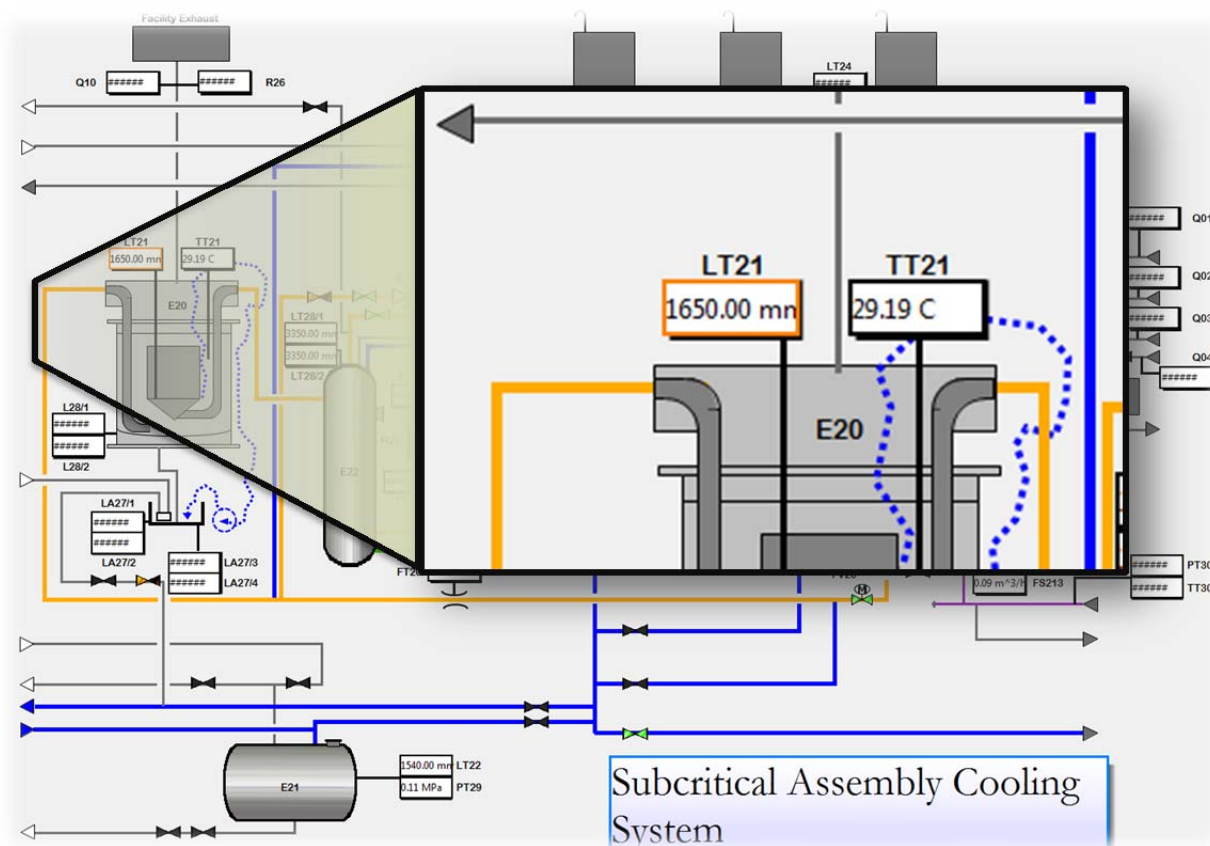


Figure 24: An example of a sensor's border changing color based upon an alarm or trip criteria.

These Alarm and Trip features are an advanced feature of the Plant Model, where NSFS can be ran without engaging the Alarm and Trip features by selecting the Disable PPS button on the PCS Control Panel, as shown in Figure 25.

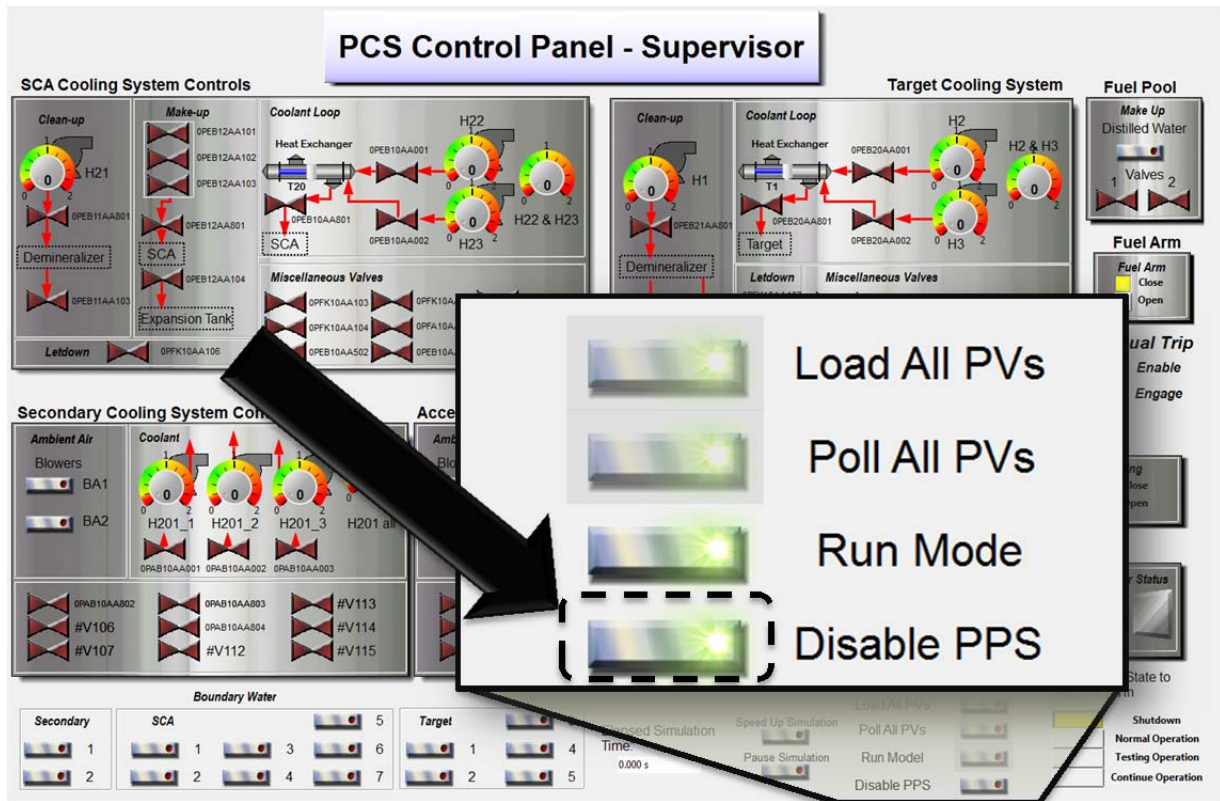


Figure 25: A close-up view of the Disable PPS button on the PCS Control Panel for the Supervisor. The Disable PPS button is in a similar position on the PCS Control Panel for the Operator.

When the *Disable PPS* option is enabled, the Alarm and Trip values will still highlight in the *Alarm Panel* as “yellow” or “red,” but the system will not engage in the Plant Protection System’s shutdown sequence. This functionality could then be used to see if the operator can bring the system to a functional state without the aid of the predefined shutdown sequence.

Refreshing the OPI Screen

There will be points through the normal operation of NSFS where the OPI screens in CSS need to be refreshed. This occurs most frequently at points when the Plant Model has been restarted and the CSS screen still has elements in its history from the remaining run of the Plant Model. A common example would be on the Plots panel, where the last n points are plotted and within that set some previous values may exist, as shown in Figure 26. By “right-clicking” on the OPI screen, the user can select the “Refresh OPI” option to force the screen to be redrawn. This will allow the screen to “forget” any remaining data from a past run of the Plant Model. Alternatively, the user may push the [F5] key as a shortcut for the “Refresh OPI” operation.

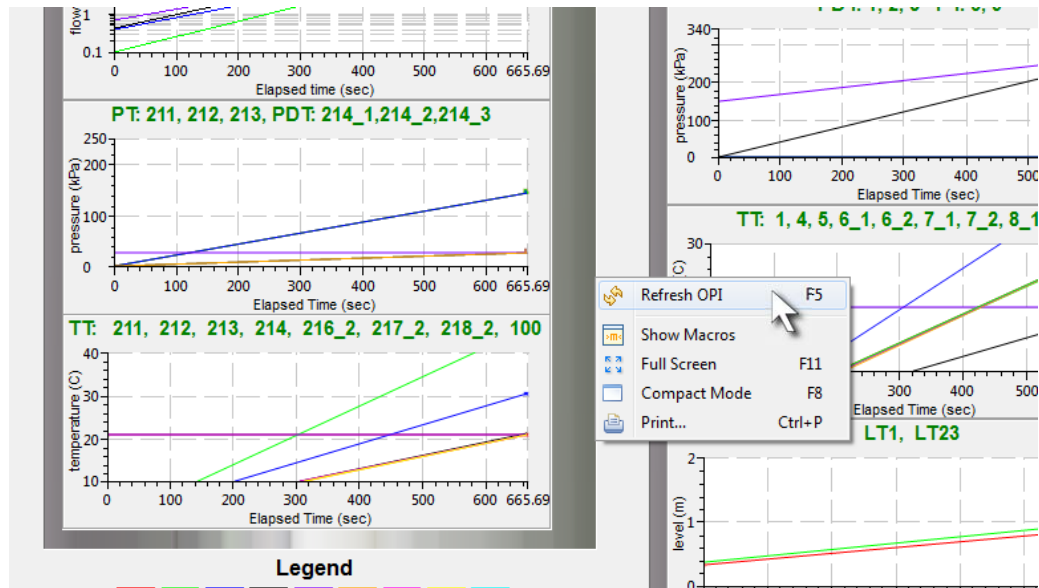


Figure 26: A situation where the Plots screen has values from a previous run still stored. Right clicking on the screen and selecting “Refresh OPI” (or pressing the F5 key) will alleviate the issue.

Exiting NSFS:

To fully exit NSFS, it is important that all three components of NSFS are properly closed, namely: EPICS, the Plant Model, and any CSS panels. If all three of these elements are not exited properly, then NSFS may continue to consume system resources after the user is finished. The individual components of NSFS consume ample amounts of computational resources (CPU time for the Plant Model, RAM for the CSS panels) and it is not suggested to leave any portion running inadvertently.

When using the Combined Operating Environment, the Python-based GUI program will attempt to close all three elements itself. In many cases, it will close EPICS and the Plant Model, but will be unable to close all of the CSS panels and the user will have to close these on their own. Additionally, it would be wise to check that the constituent elements of EPICS and the Python-based Plant Model are also no longer running on the system.

If running each of the three components in a separate environment, it is recommended that the user check to ensure that the tasks are no longer running in the background. Inadvertently leaving a task running will affect the next time the program is started, and may consume valuable computational resources in the time in between.

This page intentionally left blank.

Appendix I: Screenshots

The following pages comprise the Appendix, which contains a screenshot of every panel in NSFS for reference.

In each case, the system's loops are color-coded based on the following scheme:

- Orange → Primary Coolant Line
- Purple → Secondary Coolant Line
- Green → Demineralizer Line
- Blue → Drain Lines

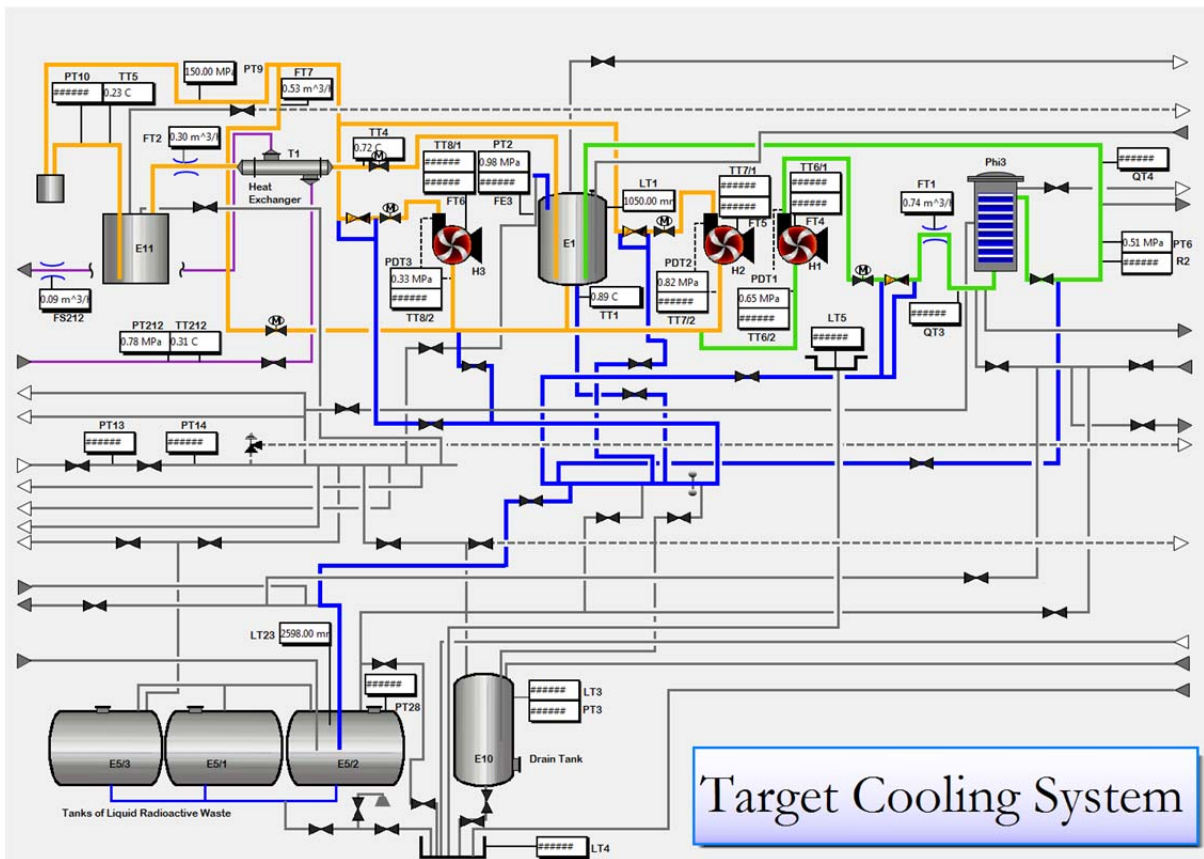


Figure 27: Target Cooling System graphical display where the coolant line is in orange, the secondary coolant line is purple, the demineralizing line in green, and the drain tank line in blue.

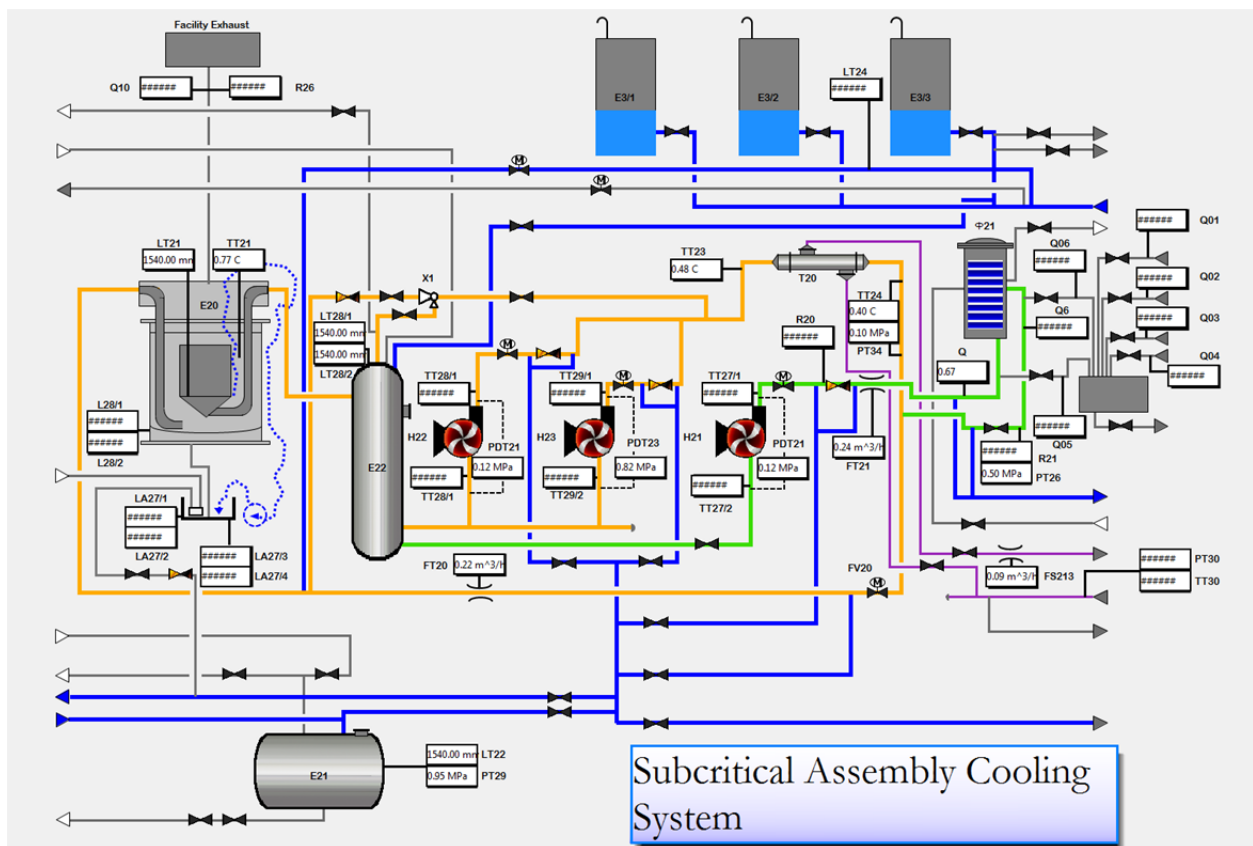


Figure 28: Subcritical Assembly Cooling System graphical display where the coolant line is in orange, the secondary coolant line is purple, the demineralizing line in green, and the drain tank line in blue.



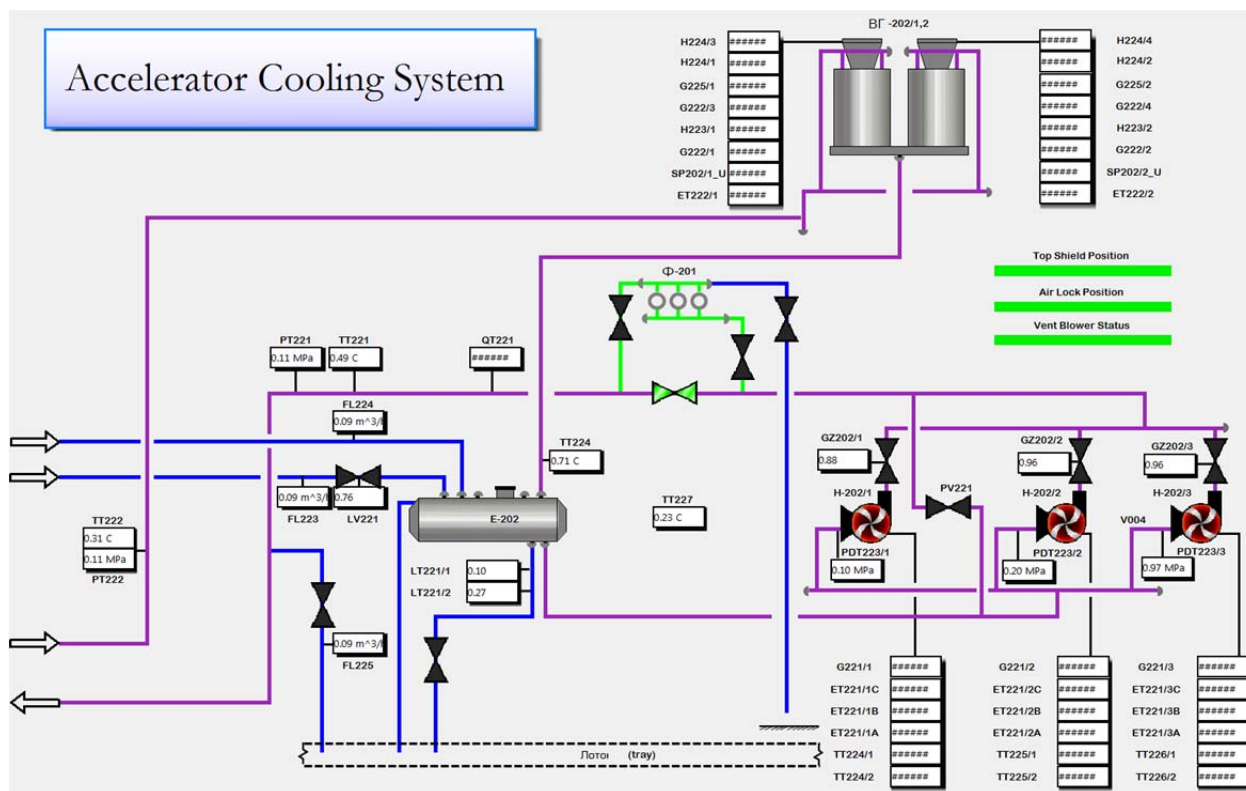


Figure 30: Accelerator Secondary Cooling System graphical display, where the primary coolant line is orange, the secondary coolant line is purple, the demineralizing line is green, and the drain line is blue.

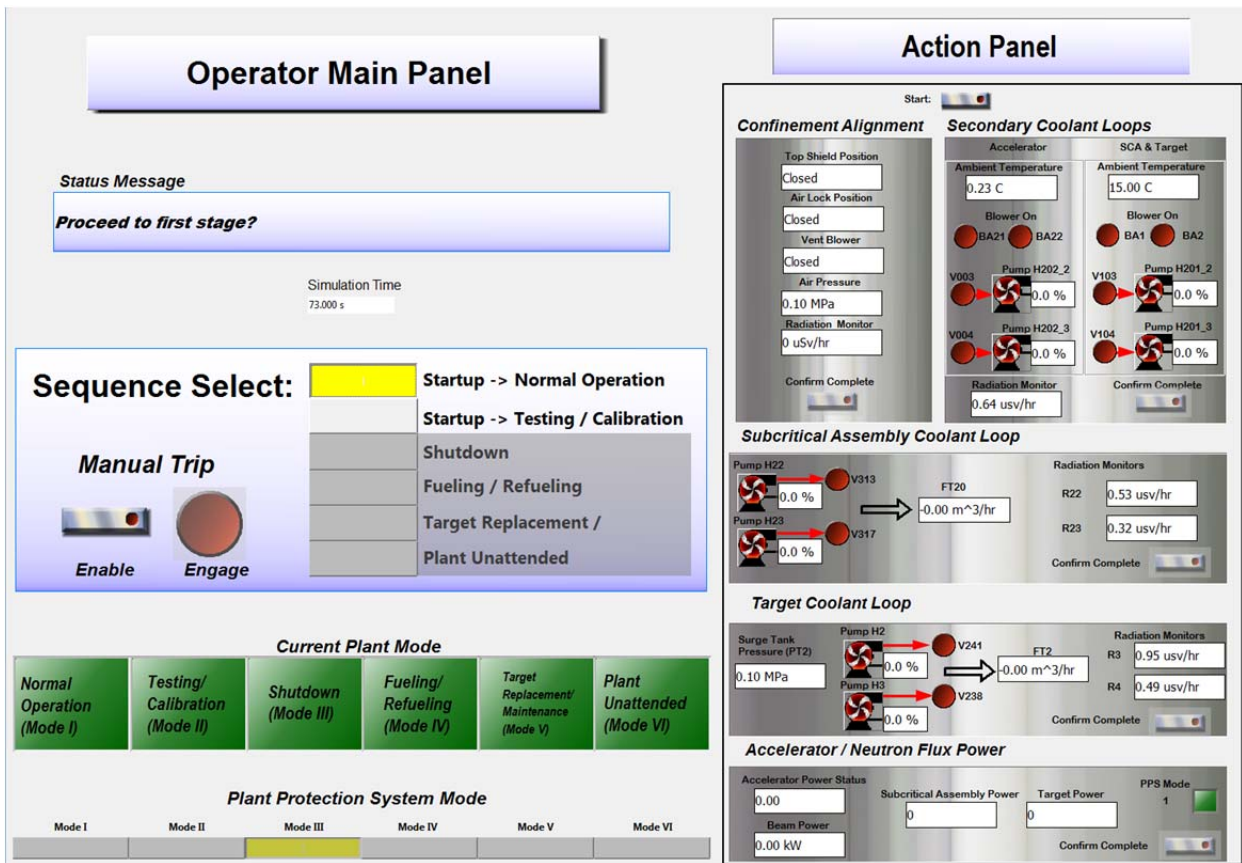


Figure 31: Operator Main Panel, where the sequence of actions that must be performed are indicated in the right section of the panel.

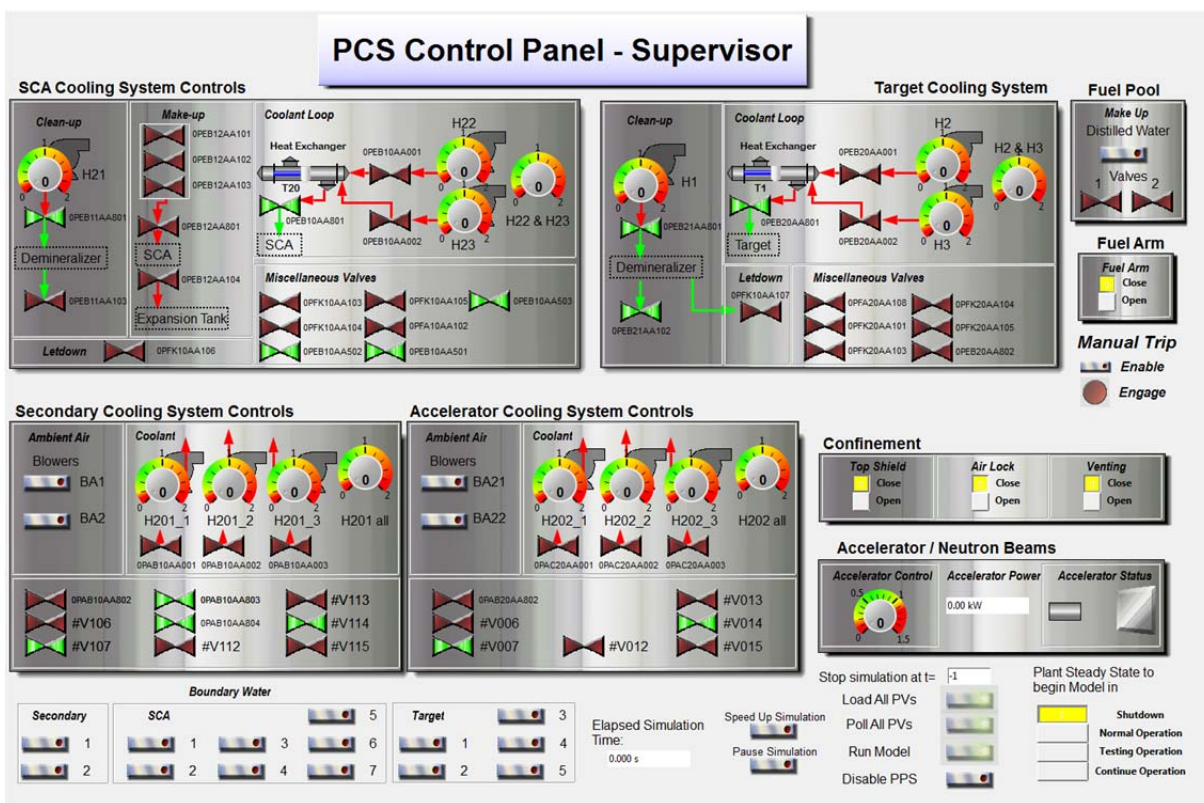


Figure 32: Plant Control System (PCS) Pane from the Supervisor point of view, where every individual component can be controlled and the Plant Model can be initiated.

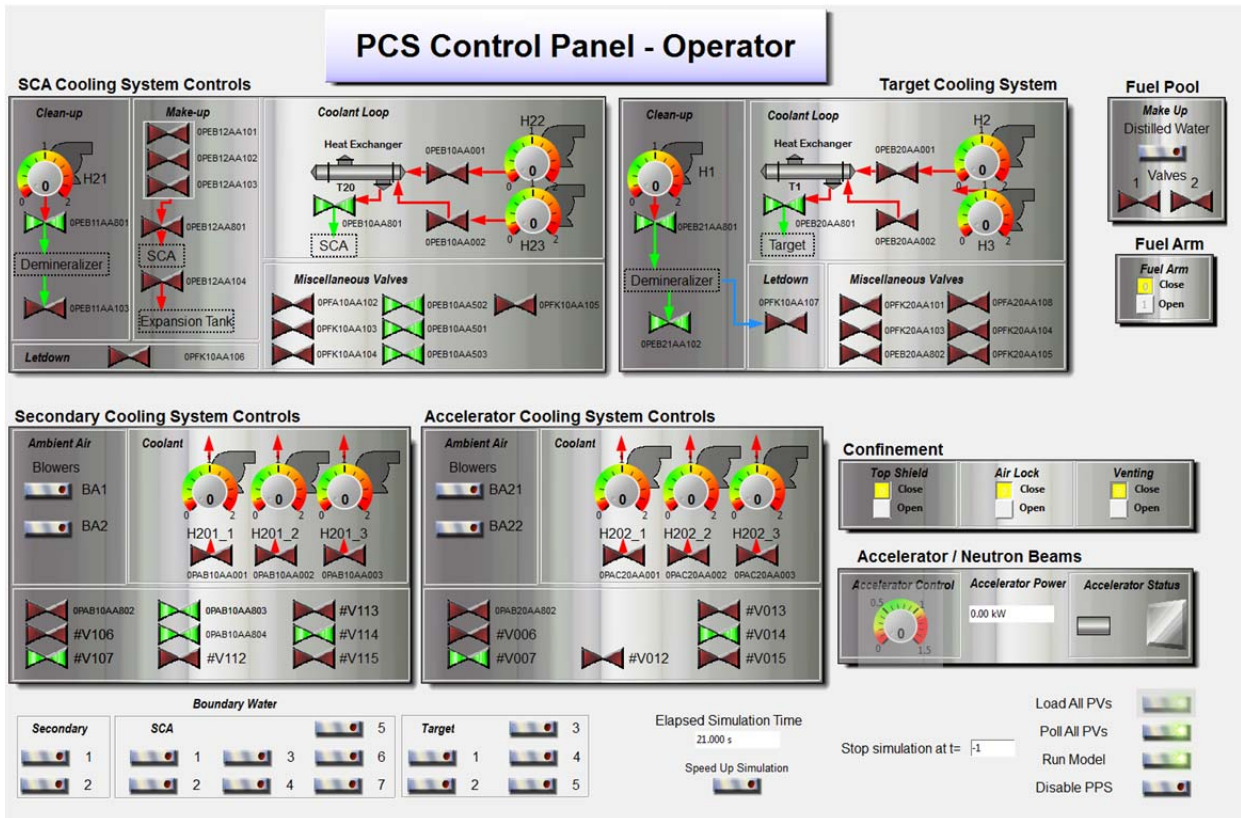


Figure 33: Plant Control System (PCS) Panel from the Operator point of view, where some components can be controlled by the operator based on the current procedure and mode.

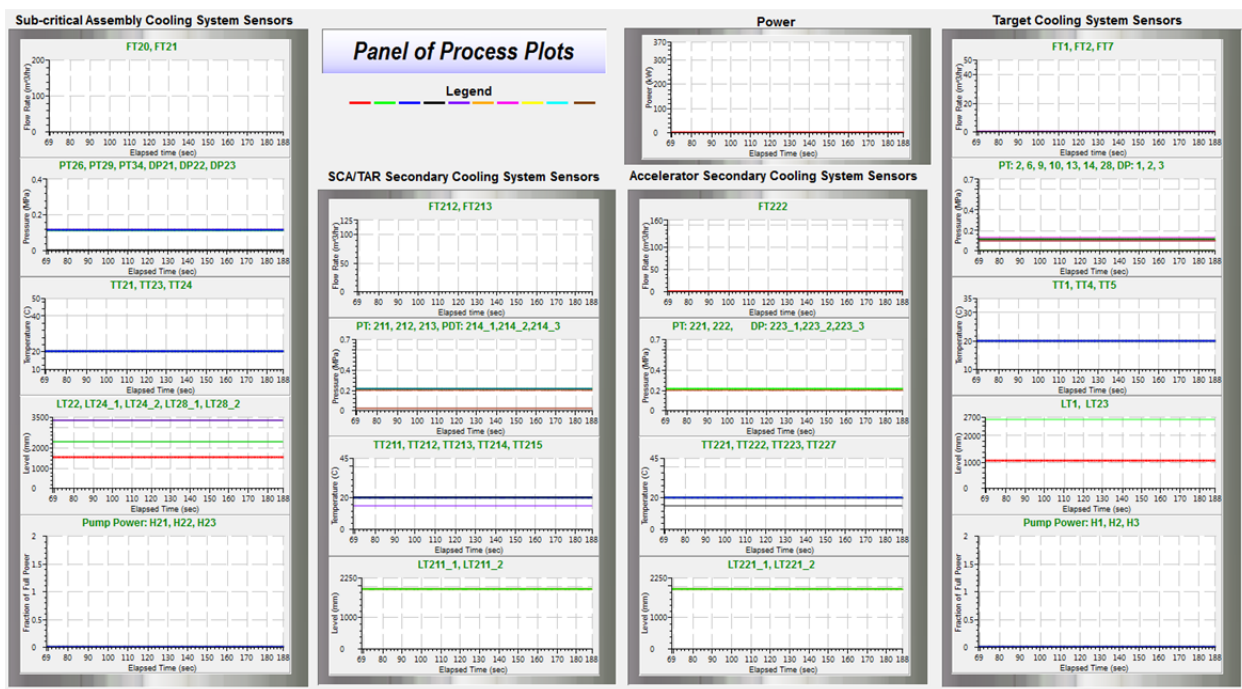


Figure 34: Panel containing the various plots of plant components and sensor values with respect to time.

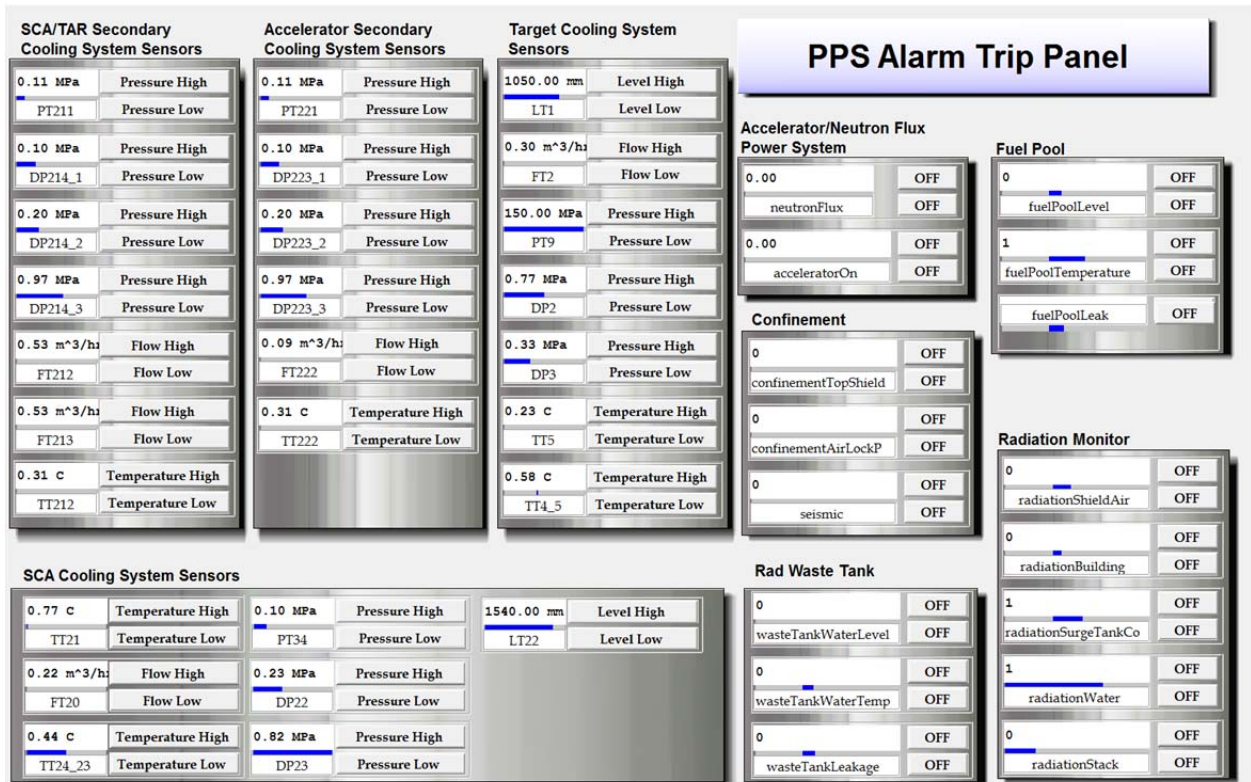


Figure 35: Panel containing the various sensors with alarms connected to them. The components change color depending on the Alarm and Trip status of the components in the current PPS mode.

ACKNOWLEDGMENT

This project is supported by the U.S. Department of Energy, Office of Global Nuclear Material Threat Reduction (NA213), National Nuclear Security Administration of US Department of Energy.



Nuclear Engineering Division

Argonne National Laboratory

9700 South Cass Avenue, Bldg. 208

Argonne, IL 60439

www.anl.gov



Argonne National Laboratory is a U.S. Department of Energy
laboratory managed by UChicago Argonne, LLC